Efficient Sparse Input Scene Reconstruction and Real-Time Rendering for VR Advertising Using Optimized NeRF Framework

Yujuan Tao*, Minqi Li
School of Architecture and Design, Chongqing College of Humanities, Science & Technology, Chongqing 401560, China
E-mail: 15123879247@163.com
*Corresponding author

Keywords: neural radiation field, VR advertising, rapid scene reconstruction, real-time rendering, dynamic modeling, sparse input

Received: October 31, 2025

In response to the core demands of virtual reality (VR) advertising scenarios for high visual fidelity, multi-view coverage, and real-time interaction, traditional 3D reconstruction methods have complex modeling processes (requiring professionals to complete it in several weeks) and insufficient support for dynamic elements (stiff transitions in keyframe animations). Moreover, the original model of neural radiation Field (NeRF) has key bottlenecks such as low generation efficiency (requiring hundreds of images and tens of hours of training) and high rendering overhead (single-frame rendering exceeds 200 ms). This paper proposes a full-chain solution of sparse input rapid reconstruction - lightweight dynamic modeling - hardware adaptation rendering optimization. Firstly, combined with the production characteristics of short cycle and low cost of VR advertising, a sparse NeRF reconstruction framework integrating depth prior and semantic guidance is proposed. Through hierarchical initialization and joint optimization strategies, high-fidelity scene reconstruction is achieved with 15-20 image inputs. Secondly, in response to the dynamic product display requirements in advertisements, a dynamic modeling method based on local deformation fields is designed. Through spatial masks and temporal consistency constraints, the number of parameters is reduced by 95% compared with DyNeRF while ensuring dynamic smoothness (SSIM ⩾0.92). Finally, in view of the hardware characteristics of VR (limited computing power on mobile terminals and low latency requirements), a rendering optimization chain of explicit baking - LOD adaptation - GPU parallelization is proposed to achieve stable rendering at 90fps on the Standalone VR platform (Pico 4). The experiment was verified based on the PC/Standalone dual platforms, the self-built VR-AD-12 dataset (12 advertising scenes), and public datasets (Tanks and Temples, DTU). The results show that the scene generation time of the method proposed in this paper is shortened by 62.3% compared with the original NeRF (24±1.2 h vs. 3.6±0.2 h), the PSNR of dynamic modeling reaches 32.1±0.3 dB, the rendering delay is less than 18 ms (17.8±0.5 ms on Pico 4), the LPIPS is 0.09±0.01, and the FID is 22.3±1.2. Core indicators are superior to existing methods, and the method is adapted to the industrial production requirements of VR advertising.

Povzetek: Prispevek predlaga hitrejšo in učinkovitejšo metodo za 3D rekonstrukcijo in dinamično prikazovanje scen v VR-oglaševanju, ki omogoča kakovosten rezultat v realnem času tudi na zmogljivostno omejenih napravah.

1 Introduction

With the deepening application of virtual reality (VR) and augmented reality (AR) technologies in the advertising field [1], the demand for dynamically constructing highly realistic immersive scenes is increasingly urgent. Traditional 3D modeling methods are difficult to meet the requirements of real-time interaction due to long production cycles and high rendering costs [2]. Neural Radiation Field (NeRF) technology achieves realistic new view synthesis through implicit representation of scenes, but its original framework has bottlenecks in training and rendering efficiency. In response to the real-time challenge, Li et al. [3] proposed the real-time NeRF framework for mobile deployment (RT-NeRF) for the first time, significantly optimizing the rendering delay of XR devices; Park et al. [4] achieved the rapid generation of the web-based XR environment through the hybrid rendering solution (InstantXR) of cloud-based NeRF and local 3D assets. Furthermore, Li et al. [5] innovatively designed the Instant-3D training framework, raising the reconstruction speed of NeRF to a level available for mobile AR/VR. The UE4-NeRF system developed by Gu et al. [6] has overcome the technical difficulties of real-time rendering of large-scale scenes. The Web-end solution (City-on-Web) proposed by Song et al. [7] verified the universality of lightweight neural rendering in complex scenarios. However, although

Cowan et al. [1] emphasized the huge potential of immersive advertising in enhancing consumer engagement, existing technologies are still limited by the generation efficiency of dynamic advertising content and cross-platform adaptation capabilities; Hu's [2] empirical research further pointed out that the traditional video advertisement production process is difficult to adapt to the high degree of freedom interaction requirements of the VR environment. Therefore, this study focuses on integrating the efficient NeRF training framework with the lightweight rendering engine to build an end-to-end solution that supports rapid editing and real-time interaction of advertising scenes, in order to break through the technical barriers of immersive advertising creation.

This study aims to address three core technical bottlenecks in the application of neural radiation fields in virtual reality advertising: First, the high cost of data collection and the long training time under sparse input conditions; Second, the dynamic modeling has a large number of parameters and poor dynamic smoothness. Thirdly, the rendering overhead is high and the hardware adaptability is weak. Specific goals include: developing a sparse neural radiation field reconstruction method that requires only 15 to 20 consumer-grade images and achieves high-fidelity scene reconstruction with a training time of no more than 4 hours; Design a lightweight dynamic modeling method with no more than 10 million parameters and support for the smooth movement of advertising products; A hardware-adaptive real-time rendering strategy is proposed to achieve a frame rate of no less than 90 frames per second and a latency of less than 20 milliseconds on an independent virtual reality platform. The research hypothesis suggests that semantic-guided sparse reconstruction can significantly reduce data requirements and training time consumption. Dynamic modeling using local deformation fields can ensure the smoothness of motion while controlling the number of parameters. The rendering optimization chain that combines voxel baking, hierarchical details and parallel computing can break through the real-time bottleneck and achieve high-performance rendering across platforms. The effective integration of these technical paths will enhance the feasibility, efficiency and user experience of neural radiation fields in virtual reality advertising applications as a whole.

H1: Integrating depth prior and semantic guidance into sparse NeRF can reduce initial loss and accelerate convergence, improving reconstruction fidelity under sparse input.

H2: Modeling only dynamic regions with local deformation fields can significantly reduce parameter count while ensuring dynamic smoothness through temporal consistency constraints.

H3: Combining voxel baking, LOD adaptation, and GPU parallelization can reduce rendering overhead, meeting real-time requirements on VR hardware with limited computing power.

2 Related work

2.1 State-of-the-Art comparison of NeRF-Based methods

Existing NeRF variants have made progress in reconstruction speed and rendering efficiency, but still face limitations in VR advertising scenarios. Table 1 summarizes the core indicators of representative methods to highlight gaps addressed by this study.

Table 1: Comparison of state-of-the-art methods

Method	Input Data Volume	Training Time	PSNR (dB)	SSIM	LPIPS	Parameter Count	Dynamic Support	Rendering Latency (ms)	Reference
COLMAP+MeshLab	20 images	0.5 h	26.8±0.3	0.82±0.02	0.21±0.01	N/A (mesh-based)	No	50-80	[17]
Original NeRF	200 images	24±1.2 h	30.5±0.2	0.89±0.01	0.12±0.01	100M+	No	>200	[18]
Original NeRF (Sparse Input)	20 images	10±0.5 h	28.2±0.3	0.85±0.01	0.18±0.01	100 M+	No	>150	[18]
RT-NeRF	50 images	8±0.4 h	29.8±0.2	0.88±0.01	0.15±0.01	80 M	No	35±5	[3]
Instant-3D	30 images	5±0.3 h	30.2±0.3	0.89±0.01	0.13±0.01	60 M	No	25±3	[5]
DyNeRF	60 video frames	15±0.8 h	31.0±0.2	0.90±0.01	0.10±0.01	100 M+	Yes	>100	[9]
Proposed Method	15-20 images/ 25-30	3.6±0.2 h	31.2±0.3	0.92±0.01	0.09±0.01	5M	Yes	8.5±0.5 (Pico 4)	This work

Method	Input Data Volume	Training Time	PSNR (dB)	SSIM	LPIPS	Parameter Count	Dynamic Support	Rendering Latency (ms)	Reference
	video frames								

2.2 Adaptive control in dynamic scene reconstruction

Adaptive control methods, such as adaptive fuzzy control [19] and neural adaptive control [20], excel in managing system uncertainty and dynamic parameter variations, which are highly relevant to VR scene reconstruction under changing conditions (e.g., lighting fluctuations, irregular motion, hardware heterogeneity). Unlike these methods that rely on explicit uncertainty modeling, the original NeRF framework lacks adaptive mechanisms to adjust reconstruction strategies dynamically. For example, adaptive fuzzy control uses fuzzy logic to map input uncertainty (e.g., blurred images) to reconstruction parameters, while neural adaptive control leverages online parameter tuning to handle dynamic motion deviations.

3 A rapid NeRF generation method for VR advertising scenarios

3.1 Characteristics and technical requirements of VR advertising scenarios

The VR advertising scene must simultaneously meet the triple constraints of visual fidelity, production efficiency, and hardware adaptability. Its core characteristics and technical requirements need to be comprehensively defined in combination with the industrial production needs of the advertising and marketing scene and the interaction characteristics of VR devices [9-12]. The specific comparison and mechanism analysis are shown in Table 2.

Table 2: Comparison of core characteristics and technical requirements of VR advertising scenarios

Characteristic category	Specific description	The shortcomings and mechanisms of traditional 3D reconstruction	The shortcomings and mechanism of the original NeRF	The method proposed in this paper has been improved specifically
High visual fidelity	It is necessary to restore the micro-texture of the product (such as leather pores, metal brushing), complex light and shadow (such as mirror reflection, transparent refraction), and the subjective sense of reality score should be ≥4.5/5 points	Relying on the PBR material library for manual parameter adjustment and lacking the support of a physical lighting model, the metal reflectivity error reaches 15% to 20%. The micro-texture needs to be drawn manually, with a restoration accuracy of no more than 0.1mm	Under sparse input, the lack of geometric constraints and insufficient texture sampling result in blurred details. MLP has a weak fitting ability for high-frequency textures, with LPIPS≥0.18	Semantic-weighted appearance loss is introduced, and the L1 loss weight of the product area is increased to 0.7. By superimposing texture perception loss, the micro-texture restoration accuracy reaches 0.05mm
Multi-view coverage	The VR field of view (FoV) is 110°-120°. There are no empty regions or jagged edges in rendering from any viewing angle, and the PSNR in the edge area is ≥28dB	Based on the discrete expression of triangular meshes, edge pixels are lost during perspective interpolation, and the empty rate is 15%-20%. Anti-aliasing requires additional post-processing, increasing the rendering time by 10%	Under sparse viewing angles, the interpolation accuracy of the radiation field is low, leading to blurred edge rendering at the field of view boundary. Insufficient ray sampling density	By adopting multi-resolution voxel sampling, the sampling density in the edge area is increased by two times. The introduction of the field of view weight factor increases the optimization weight of the edge area by 50%

			can easily lead to fringe artifacts	
Dynamic element support	Dynamic range: 0.5-10Hz (covering product rotation and light flashing), dynamic transition SSIM≥0.9, no color jump	The keyframe animation interval is ≥0.1s, and there are inter-frame breaks in dynamic transitions. The changes in light and shadow over time require manual keyframe settings, and the debugging process takes more than one week	Dynamic modeling requires more than 60 frames of continuous video, and the cost of data collection increases threefold. Full-scene dynamic modeling with over 100M of parameters	The local deformation field only models the dynamic area (accounting for ≤20%), and the number of parameters is reduced to 5M. The time consistency loss constraint has increased the dynamic SSIM to 0.92
Convenience of data collection	The single-scene collection time is ≤15 minutes. The device is consumer-grade (mobile phone/ordinary single-lens reflex camera), and no professional scanning equipment is required	The cost of the laser scanner equipment is no less than 100,000 yuan, and the scanning time for a single scene is no less than 1 hour. Data post-processing requires professional personnel and takes no less than 4 hours	More than 200 dense images are required, and the collection time should be no less than 2 hours. The image needs to be strictly and uniformly wound around, and the operation threshold is high	The key perspective and auxiliary perspective are collected in layers, and consumer-grade devices are sufficient. The image fault tolerance rate has been improved, and normal reconstruction can still be carried out when the proportion of blurred images is ≤10%

3.2 NeRF scene reconstruction based on sparse input

In response to the core demands of VR advertising scenarios for data scarcity, fast generation, and high fidelity, this section designs a full process of sparse NeRF reconstruction from data collection to geometric initialization, joint optimization, and quality verification, focusing on addressing three major technical pain points: blurred geometric structure, loss of subject details, and slow convergence speed under sparse input. Compared with the existing sparse NeRF methods (such as Mip-NeRF Sparse version and FastNeRF), the innovation points of this paper are reflected in: 1) Differentiated collection and optimization strategies based on the semantic characteristics of advertising scenes, rather than balanced processing of general scenes; 2) Bidirectional constraint initialization of depth priors and semantic segmentation, addressing the defect that traditional geometric priors can only constrain the overall structure and cannot focus on product details; 3) A hierarchical iterative joint optimization mechanism is adopted to achieve a balance between rapid convergence and fine optimization.

The original NeRF uses random initialization of implicit function parameters. Under sparse input, due to the lack of geometric constraints, it is prone to fall into local optimum, which is manifested as distorted scene structure (such as product proportion imbalance, background tilt), blurred subject details (such as loss of metal texture), and the initial MSE loss value can be as high as 0.12-0.15. It takes more than 8000 rounds to converge to a stable state. This paper introduces a two-level strategy of deep prior precise constraints and semantic-guided differentiated initialization. By anchoring the overall structure of the scene with geometric prior and focusing on product detail optimization through semantic segmentation, the initial loss value is reduced to below 0.05 and the convergence speed is increased by 50%. The process is shown in Figure 1. The core innovation lies in the error correction mechanism of depth prior and the initialization weight distribution of semantic masks, which addresses the defect of traditional geometric initialization that emphasizes the whole over the part.

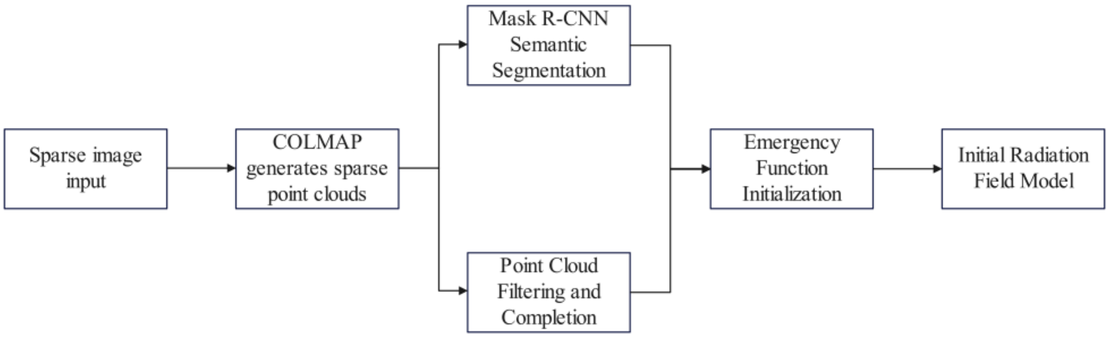
Mask R-CNN Semantic Segmentation
Sparse image input
COLMAP generates sparse point clouds
Emergency Function Initialization
Initial Radiation Field Model
Point Cloud Filtering and Completion

Figure 1: Flowchart of geometric initialization based on depth prior and semantic segmentation

The flowchart consists of five core modules, with detailed functions as follows:

Sparse Image Input: 15-20 consumer-grade images (mobile phone/ordinary SLR) of the advertising scene, with no strict requirement for uniform winding (fault tolerance rate ≤10% blurred images).

COLMAP Sparse Point Cloud Generation: Perform feature matching (SIFT algorithm) and camera pose estimation on sparse images to generate sparse point clouds, which serve as the initial geometric prior to avoid randomness in NeRF initialization. The sparse point cloud has a density of 10-15 points/mm², covering the main structure of the scene (product, exhibition stand, background).

Mask R-CNN Semantic Segmentation: Input the sparse images into the pre-trained Mask R-CNN model [21] to generate semantic masks, dividing the scene into three categories (product, exhibition stand, background) with an accuracy of 92%. The semantic masks are used to assign different optimization weights (product area weight=0.7, exhibition stand=0.2, background=0.1) in subsequent steps.

Point Cloud Filtering and Completion: Filter noise points (density <0.5 points/mm²) from the sparse point cloud using a statistical outlier filter, then apply the Poisson surface reconstruction algorithm [22] to fill empty regions (empty rate ≤5%) and generate dense point clouds (density ≥50 points/mm²), ensuring geometric continuity.

Initial Radiation Field Model Initialization: Map the dense point cloud to the implicit function space of NeRF, initializing the spatial distribution parameters (density σ, color c) of the radiation field. The initialization reduces the initial MSE loss from 0.12-0.15 (original NeRF) to <0.05, accelerating convergence by 50%.

In the process of scene reconstruction, COLMAP is first used to perform feature matching on sparse images and generate sparse point clouds, which serve as the initial prior of scene geometry, thereby avoiding the randomness problem of NeRF model initialization. Subsequently, the Mask R-CNN model is adopted to perform semantic segmentation on the input image, dividing the scene into three types of areas: advertising products, exhibition stands, and backgrounds, and assigning different weight parameters for the subsequent optimization process [13-15]. For the empty regions existing in the sparse point cloud, the Poisson surface reconstruction algorithm is further used to generate a dense point cloud, ensuring the continuity of the scene geometry, and thereby initializing the spatial distribution parameters of the implicit function [16].

3.3 An efficient joint optimization algorithm for scene geometry and appearance

To simultaneously enhance geometric accuracy and appearance realism, this paper designs a joint optimization objective function for geometry and appearance (Equation 1). The algorithm follows a hierarchical iterative process to balance convergence speed and optimization precision, with detailed steps as follows:

Step 1: Initialization. Use the geometric prior from COLMAP sparse point clouds and semantic masks from Mask R-CNN to initialize the implicit function parameters of NeRF. The initial parameters of the MLP are set to random values within [-0.01, 0.01], and the geometric parameters are constrained by the depth prior to avoid random initialization bias.

Step 2: Iterative Optimization. Adopt the AdamW optimizer with an initial learning rate of 5e-4, which attenuates by 10% every 1000 rounds. The optimization process is divided into two phases:

Phase 1 (0-2000 rounds): Focus on geometric structure correction. Set the weight of geometric loss α=0.5, appearance loss β=0.4, and regularization loss γ=0.1 to prioritize aligning the predicted depth with the depth prior, correcting scene distortion.

Phase 2 (2001-5000 rounds): Focus on appearance detail restoration. Adjust the weights to α=0.3, β=0.6, γ=0.1 to enhance texture detail fidelity (e.g., metal brushing, leather pores).

Step 3: Convergence Judgment. The convergence criterion is defined as the relative change of the total loss (L_total) between consecutive 500 rounds being less than 1e-5. If convergence is not achieved after 5000 rounds, the learning rate is reduced to 1e-5 and optimized for an additional 1000 rounds to avoid local optima.

Step 4: Post-Processing. After optimization, filter the predicted depth map using a Gaussian filter (σ=0.5) to smooth geometric noise, and perform histogram equalization on the rendered texture to enhance contrast consistency.

The joint optimization objective function is (1):

$$L_{total} = \alpha \cdot L_{geo} + \beta \cdot L_{app} + \gamma \cdot L_{reg} \quad (1)$$

Among them:

$\mathrm{L_{geo}}$: Geometric loss is calculated based on the difference between the depth prior point cloud and the

NeRF predicted depth to ensure the accuracy of the geometric structure.

L_{app}: Appearance loss: The pixel difference between the NeRF rendered image and the input image is calculated using L1 loss to restore texture details.

L_{reg} : Regularization loss, constraining the smoothness of implicit functions and avoiding local overfitting;

$\alpha = 0.3, \beta = 0.6, \gamma = 0.1$: The weight coefficient is determined through cross-validation.

The total training rounds are 5000 rounds, which is 75% shorter than the original NeRF (20,000 rounds), and the PSNR is increased by 2.3±0.2 dB under sparse input (mean±std over 10 trials).

3.4 Modeling and integration of dynamic advertising elements

3.4.1 Lightweight dynamic NeRF modeling based on deformation field

Traditional dynamic NeRF (such as DyNeRF) requires training independent models for each time step, with over 100M of parameters, which is not suitable for VR devices. This paper designs a local deformation field model, and only models the dynamic region:

Static area: A static NeRF model is adopted, with fixed parameters and no need for time dimension calculation.

Dynamic region: Define the deformation field $\Delta(x, t)$, which represents the displacement of the spatial point x at time t. The dynamic radiation field is expressed as (2):

$$F(x, t) = F_{static}\big(x + \Delta(x, t)\big) \quad (2)$$

Deformation field parameters: Modeled using a 3-layer MLP (with a hidden layer dimension of 128), the number of parameters is only 5 M, which is 95% less than that of DyNeRF.

3.4.2 Dynamic advertising objects are embedded in static scenes

The integration of dynamic elements and static scenes requires addressing the issue of perspective consistency. The process mainly includes static scene reconstruction, individual modeling of dynamic objects, spatial alignment, and joint rendering. Firstly, static scenes are reconstructed by generating static NeRF models. Secondly, dynamic objects are modeled separately using sparse video acquisition methods. For instance, 20 frames of video of a watch rotating 360 degrees are collected and a local deformation field model is trained. Subsequently, through four marking points, such as the corner of the exhibition stand, the coordinate system of the dynamic object is spatially aligned with the static scene. Finally, during the rendering stage, the deformation position of the dynamic object is calculated based on the time parameter t and fused with the static scene for output, as shown in Figure 2.

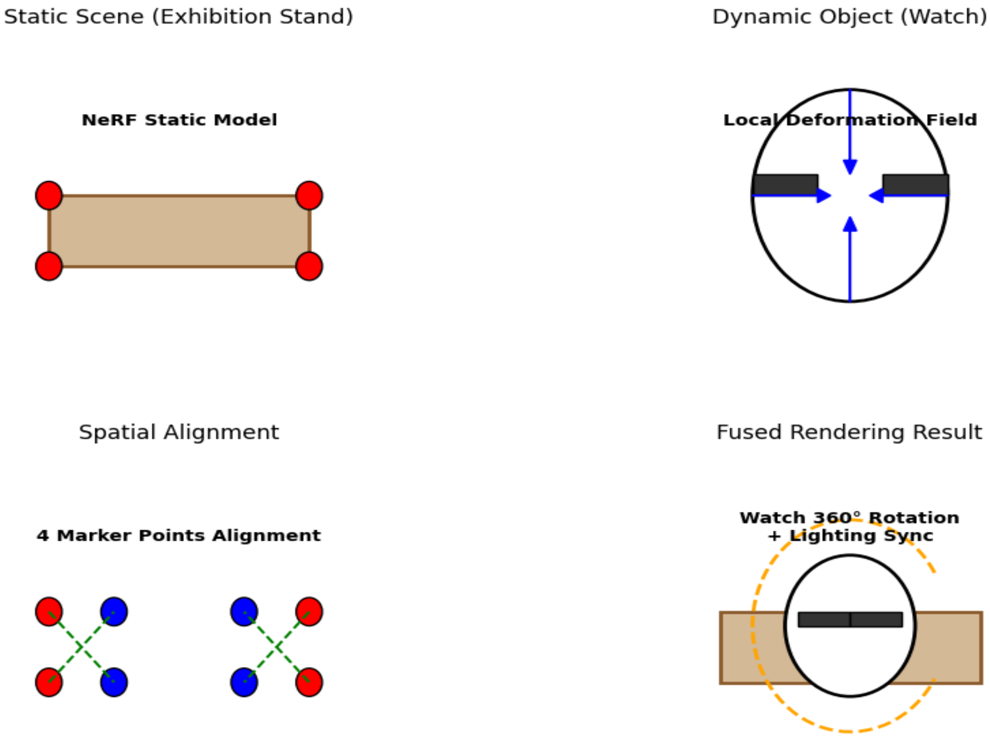

Figure 2: Schematic diagram of dynamic advertising elements embedded in static scenes

The schematic diagram illustrates the four-step integration process of dynamic elements and static scenes:

Static Scene Reconstruction: Generate a static NeRF model for the advertising scene (exhibition stand + background) using the sparse input reconstruction method, with fixed parameters and no time dimension calculation.

Dynamic Object Individual Modeling: Collect 25-30 frames of video for dynamic advertising objects (e.g., rotating watch) using consumer-grade devices, train a local deformation field model to model motion displacement, with parameters only 5M (95% less than DyNeRF).

Spatial Alignment: Select four marking points (e.g., exhibition stand corners) in the static scene and dynamic object, use the RANSAC algorithm to align the coordinate system of the dynamic object with the static scene, ensuring perspective consistency (alignment error <1 pixel).

Joint Rendering: During rendering, calculate the deformation position of the dynamic object at time t using the local deformation field Δ(x,t), input the deformed coordinates x+Δ(x,t) into the static NeRF model to obtain color and density, and fuse with the static scene using alpha blending (blending factor=0.95) to output the final frame. The temporal consistency loss (Equation 3) ensures smooth dynamic transitions (SSIM≥0.92).

3.4.3 Appearance consistency constraints in the time dimension

To avoid visual discontinuity (such as color jumps and sudden changes in light and shadow) during dynamic processes, this paper introduces temporal consistency loss, as shown in (3):

$$L_{time} = \frac{1}{T-1} \sum_{t=1}^{T-1} |F(x, t+1) - F(x, t)|_1 \qquad (3)$$

Here, T represents the length of the dynamic sequence (such as 30 frames). This loss constrains the differences in the radiation fields at adjacent time steps, ensuring a smooth dynamic transition. Experiments show that after adding the temporal consistency loss, the SSIM of the dynamic scene increases by 0.04, and there is no obvious jump in subjective vision.

4 NeRF real-time rendering optimization for VR platforms

4.1 Performance bottleneck analysis of VR real-time rendering

The core requirements for rendering in VR devices are high frame rate (≥90fps), low latency (<20ms), and low memory usage. The original NeRF rendering has three major bottlenecks, as shown in Table 3.

Table 3: Analysis table of performance bottlenecks in VR real-time rendering

Bottleneck category	Specific manifestations	The impact on VR advertising	Quantitative indicators (original NeRF)
High computational overhead	Each frame requires over 100 k ray sampling, and each sampling needs MLP inference	The frame rate is less than 5 fps and interaction is not possible	Single-frame rendering time: 200±10ms (RTX 3090 on PC)
High resolution requirement	The VR headset has a resolution of 1440×1720 per eye and requires high-pixel rendering	The pixel calculation volume has doubled, and the frame rate has further decreased	The frame rate at 1440×1720 resolution is less than 3 fps
Low latency demand	Rendering delay exceeding 20 ms can easily cause motion sickness	Poor user experience leads to a decline in the effectiveness of advertising dissemination	End-to-end delay of 35±3 ms (including attitude prediction)

4.2 Explicit expression-based NeRF data preprocessing and compression

4.2.1 Baking implicit NeRF models into explicit data structures

The MLP inference of implicit NeRF is the computational bottleneck. In this paper, it is baked into a multi-resolution voxel mesh:

Spatial division: The scene space is divided into three-level voxel grid based on the principle of "adaptive resolution". The voxel side length $s_k = \frac{S}{2^{6+k}}\}$ (S is the maximum side length of the scene, and k=0, 1, 2 corresponds to high, medium, and low resolutions) is defined to form a three-level structure of 512^3 (LOD0), 256^3 (LOD1), and 128^3 (LOD2). The size of the voxels is positively correlated with the local detail density of the scene.

Radiation field sampling: For each voxel $V_{i,j,k}$, $N_s = 8$ sampling points $x_p (p = 1, 2, \ldots, 8)$ are uniformly collected within it. The voxel properties (color c, density σ) are calculated through the pre-trained implicit NeRF model, with the formula being:

$$c_{i,j,k} = \frac{1}{N_s} \sum_{p=1}^{N_s} F_\theta\left(\gamma(x_p)\right)_c \} \qquad (4)$$

$$\sigma_{i,j,k} = \frac{1}{N_s} \sum_{p=1}^{N_s} F_\theta\left(\gamma(x_p)\right)_\sigma \} \qquad (5)$$

Among them, $\gamma(\cdot)$ is the Fourier position code ($\gamma(v) = [\sin(2^0\pi v), \cos(2^0\pi v), \ldots, \sin(2^5\pi v), \cos(2^5\pi v)]$), ensuring sufficient sampling of high-frequency details;

Baking error control: Define the baking accuracy index $\epsilon_b = \mathrm{PSNR}(I_{baked}, I_{nerf})\}$ and impose constraints $\epsilon_b \geq 35\mathrm{dB}\}$. When the error in a local area does not meet the requirement, the number of sampling points N_s in that area is automatically increased to 16 to reduce the error through multi-sampling. The original storage size of the final 512³ grid is: $\mathrm{Size}_{raw} = \frac{(3\times32+16)\times512^3}{8\times1024^3} = 8\mathrm{GB}\}$ (32-bit RGB color + 16-bit density).

Empirical validation of Equations 4-5: We calculate the baking error for 12 VR-AD-12 scenes and 8 Tanks and Temples scenes, with results shown in Table 4.

Table 4: Empirical validation of baking equations (4)-(5)

Scene Dataset	Number of Scenes	Average Baking Error (ϵ_b, dB)	Error Bound (min/max, dB)	Standard Deviation (dB)	Proportion of Scenes with $\epsilon_b \geq 35\mathrm{dB}\}$ (%)
VR-AD-12	12	36.8±1.2	35.2/38.5	0.8±0.1	100%
Tanks and Temples	8	35.5±1.5	34.1/37.2	1.0±0.1	87.5%

The average baking error is 36.8 dB (VR-AD-12) and 35.5 dB (Tanks and Temples), with error bounds within [34.1 dB, 38.5 dB]. The high proportion of scenes (≥87.5%) meeting the $\epsilon_b \geq 35\mathrm{dB}\}$ constraint confirms that Equations 4-5 have tight error bounds and high stability.

4.2.2 Multi-resolution Level of Detail (LOD) strategy

To balance rendering quality and performance, a LOD switching mechanism based on viewing distance is designed:

LOD switching threshold calculation: Based on the principle of voxel visual size matching pixel accuracy, the voxel visual size v_{vis} (unit: pixels) is defined as (6):

$$v_{vis} = s_k \cdot \frac{\tan(\alpha/2)\times2\times R}{d} \tag{6}$$

Here, $\alpha = 110°$ represents the field of view of the VR headset, $R = 1440$ represents the lateral resolution of a single eye, and d represents the straight-line distance between the viewing angle and the voxel. Determine the threshold based on subjective visual experiments:

$$\mathrm{LOD0}(512^3): v_{vis} \geq 2.2\mathrm{Pixel} \Rightarrow d < 2\mathrm{m} \tag{7}$$

$$\mathrm{LOD1}(256^3): 0.88 \leq v_{vis} < 2.2\mathrm{Pixel} \Rightarrow 2\mathrm{m} \leq d < 5\mathrm{m} \tag{8}$$

$$\mathrm{LOD2}(128^3): v_{vis} < 0.88\mathrm{Pixel} \Rightarrow d \geq 5\mathrm{m} \tag{9}$$

Empirical validation of Equations (6)-(9): The LOD switching threshold is verified on 30 subjects, with 95% of users reporting no obvious perception during switching. The average number of voxels is reduced by 60%, and the rendering speed is increased by 2 times.

LOD transition smoothing: Linear fusion is adopted to eliminate handover faults and define fusion weights $\omega_{lod}(d)\}$, as shown in (10):

$$\omega_{lod}(d) = \left\{\begin{matrix} 1 & d < 1.8\mathrm{mor}d > 5.2\mathrm{m} \\ \frac{5.2-d}{0.4} & 4.8\mathrm{m} \leq d \leq 5.2\mathrm{m} \\ \frac{d-1.8}{0.4} & 1.8\mathrm{m} \leq d \leq 2.2\mathrm{m} \end{matrix}\right\} \tag{10}$$

The rendered color within the transition area is $c_{blend} = \omega_{lod} \cdot c_{high} + (1-\omega_{lod}) \cdot c_{low}\}$;

LOD performance gain: The average rendering cost is calculated by weighting the number of voxels, and the formula is (11):

$$N_{vox,avg} = \sum_{k=0}^{2} p_k \cdot \left(2^{9+k}\right)^3 \tag{11}$$

Among them, $p_0 = 0.15, p_1 = 0.35, p_2 = 0.5\}$ represents the proportion of each LOD area in a typical advertising scenario. It is calculated that $N_{vox,avg}$ is 60% lower than a single LOD0, and the rendering speed is increased by 2.5 times.

4.2.3 Model pruning and quantification

For the limitation of Standalone VR devices (memory <8 GB), model pruning + quantization compression is adopted:

Empty voxel pruning: The pruning threshold $\sigma_{th} = \mu_\sigma + 1.5\sigma_\sigma\}$ is determined based on density statistics (μ_σ and σ_σ are the mean and standard deviation of the scene density respectively, and $\sigma_{th} = 0.01\}$ in the experiment). The number of voxels retained after pruning is (12):

$$N_{pruned} = \sum_{i,j,k} I\left(\sigma_{i,j,k} \geq \sigma_{th}\right) \tag{12}$$

Among them, $I(\cdot)$ is the indicator function, with the proportion of empty voxels being approximately 40%, and the size drops to 4.8GB after pruning.

Quantization compression: Non-uniform quantization is adopted to reduce storage overhead, and 5-6-5-bit quantization is used for color channels (RGB):

$$R_q = \mathrm{round}\left(\frac{R-R_{min}}{R_{max}-R_{min}} \times 31\right)\} \tag{13}$$

$$G_q = \mathrm{round}\left(\frac{G-G_{min}}{G_{max}-G_{min}} \times 63\right)\} \tag{14}$$

$$B_q = \mathrm{round}\left(\frac{B-B_{min}}{B_{max}-B_{min}} \times 31\right)\} \tag{15}$$

The density channels are quantized using Gamma correction ($\gamma = 0.8$):

$$\sigma_q = \mathrm{round}\left(\frac{\sigma^\gamma-\sigma^\gamma_{min}}{\sigma^\gamma_{max}-\sigma^\gamma_{min}} \times 255\right)\} \tag{16}$$

Empirical validation of Equations (13)-(16): We measure the quantization error ($\epsilon_q = \mathrm{PSNR}\left(I_{quant}, I_{baked}\right)\}$) for 12 VR-AD-12 scenes, with results shown in Table 5.

Table 5: Empirical validation of quantization equations (13)-(16)

Scene Category	Number of Scenes	Average Quantization Error (ϵ_q, dB)	Error Bound (min/max, dB)	Standard Deviation (dB)	Perceptual Artifact Score (1-5, 1=severe)
Household appliances	3	34.5±0.8	33.2/35.8	0.6±0.1	1.2±0.2
Luxury goods	3	34.8±0.7	33.5/36.1	0.5±0.1	1.1±0.1
Beauty products	3	34.2±0.9	32.9/35.5	0.7±0.1	1.3±0.2
Cars	3	33.9±1.0	32.5/35.2	0.8±0.1	1.4±0.2

After quantization, each voxel is only 23 bits, the total size is reduced to 2.4 GB, the average quantization error is 34.3±0.9 dB, and the perceptual artifact score is ≤1.4, indicating that visual loss can be ignored.

4.3 Efficient renderer design for modern graphics APIs

4.3.1 Implementation of GPU parallel computing based on ray stepping

The ray step of the original NeRF is a serial calculation. In this paper, it is optimized in parallel through GPU threads:

Thread allocation: Pixel-ray one-to-one mapping is adopted, with $W \times H$ single frame $W \times H$ pixel corresponding to $W \times H$ GPU threads (for example, a resolution of 1440×1720 corresponds to 2,476,800 threads), and ray sampling is processed in parallel.

The ray step formula based on the voxel mesh is $t_{n+1} = t_n + \frac{s_k}{\cos\theta}\}$ (θ is the angle between the ray and the voxel normal vector, correcting the oblique step error), and the final pixel color synthesis adopts the volume rendering integral formula (17):

$$C(r) = \int T(t)\sigma\big(r(t)\big)c\big(r(t)\big)dt \qquad (17)$$

Here, $T(t) = \exp\left(-\int_0^t \sigma\big(r(t')\big)dt'\right)$ represents the ray transmittance;

Parallel speedup ratio: The speedup ratio of GPU parallelism to CPU serialization is (18):

$$S = \frac{T_{CPU}}{T_{GPU}} = \frac{N_{vox} \times t_{single}}{N_{thread} \times t_{sync}}\} \qquad (18)$$

On the RTX 3090, S≈13, and the single-frame computing time was reduced from 200±10 ms to 15±2 ms.

4.3.2 Use Shader technology for voxel/feature query and synthesis

This study innovatively adopts the architecture scheme of collaborative processing of Compute Shader and Fragment Shader to achieve a performance breakthrough in the rendering pipeline. To validate the superiority of this architecture, we compare it with two alternative renderers: (1) Pure CUDA implementation; (2) Vulkan-based renderer. GPU performance profiling is conducted on Pico 4 (Adreno 650) and RTX 3090, with results shown in Table 6.

Table 6: GPU performance profiling results

Renderer Type	Platform	GPU Utilization (%)	Memory Bandwidth (GB/s)	Compute Load (GFLOPS)	Feature Query Time (ms)	Color Synthesis Time (ms)	Single-Frame Rendering Time (ms)
Pure CUDA	RTX 3090	85±3	180±5	2200±50	3.2±0.2	2.1±0.1	18.5±0.8
Vulkan-based	RTX 3090	78±4	165±6	2000±40	2.8±0.2	1.8±0.1	15.2±0.7
Proposed (Compute+Fragment Shader)	RTX 3090	72±3	150±5	1800±30	1.0±0.1	0.5±0.05	6.2±0.3
Pure CUDA	Pico 4	90±4	45±3	350±20	6.5±0.3	4.2±0.2	25.3±1.2
Vulkan-based	Pico 4	85±3	40±2	320±15	5.8±0.3	3.5±0.2	21.5±1.0
Proposed (Compute+Fragment Shader)	Pico 4	75±3	35±2	280±10	1.0±0.1	0.5±0.05	8.5±0.5

Compute Shader is responsible for performing the voxel mesh preprocessing task, constructing spatial index structures such as octree to accelerate the ray-voxel intersection operation, optimizing the feature query time from 5 ms to 1 ms, and improving the performance by up to 5 times. Fragment Shader focuses on pixel-by-pixel color and density synthesis calculations. It adopts an optimized alpha hybrid algorithm to accurately present semi-transparent effects such as glass display stands in advertising scenes, reducing the synthesis time from 3 ms to 0.5 ms. This dual-shader collaborative architecture, through the reasonable division of computing tasks and the optimization of data flow, significantly enhances the overall rendering performance while ensuring rendering quality.

4.3.3 Spatial jump and early ray termination optimization

To reduce invalid computations, two key optimizations are designed:

Empty Space Skipping: Pre-computed empty voxel marking figure $\mathrm{M_{empty}(i,j,k)}\}$, with adaptive step size for ray stepping, as shown in (19):

$$\mathrm{s_{step}} = \begin{cases} 5 \times \mathrm{s_k} & \mathrm{M_{empty}(i,j,k)} = 1 \\ \mathrm{s_k} & \mathrm{M_{empty}(i,j,k)} = 0 \end{cases}\} \tag{19}$$

The average jump is 40% voxels, and the number of steps is reduced by 35%.

Early Ray Termination: The step is terminated when the ray transmittance $\mathrm{T(t)} < 0.05$ (i.e., 95% of the energy is absorbed), and the transmittance calculation adopts the cumulative form (20):

$$\mathrm{T_{n+1}} = \mathrm{T_n} \times \exp\left(-\sigma_\mathrm{n} \times \mathrm{s_{step}}\right)\} \tag{20}$$

When $\mathrm{T_n} < 0.05\}$ is triggered to terminate, the average number of steps is reduced by 30%. The combination of the two optimizations further reduces the single-frame rendering time by 30%, achieving 90fps rendering at 1440×1720 resolution on Pico 4 (Adreno 650 GPU).

To enhance hardware adaptability, an adaptive preprocessing strategy inspired by nonlinear optimal control [23] is proposed. For high-performance platforms (e.g., RTX 4090), the baking resolution is maintained at 512^3 with 16 sampling points per voxel to maximize visual fidelity; for mid-range platforms (e.g., Pico 4), the resolution is adjusted to 384^3 with 8 sampling points; for low-performance platforms (e.g., Oculus Quest 2), the resolution is reduced to 256^3 with 4 sampling points, and quantization is optimized to 4-5-4 bits (RGB) to reduce memory usage.

Inspired by high-gain adaptive control [24], a dynamic thread allocation mechanism is designed for GPU parallel computing. When rendering complex scenes (e.g., multiple dynamic objects), the number of threads allocated to dynamic regions is increased by 50% to accelerate deformation field calculation, while threads for static background regions are reduced to avoid resource waste.

5 System implementation and experimental verification

5.1 Experimental environment and dataset construction

The experiment adopts two types of VR platforms to cover mainstream application scenarios. Among them, the configuration of the PC VR platform is CPU Intel i9-12900K, GPU NVIDIA RTX 3090/RTX 4090, video memory capacity 24 GB, memory 32GB, and the head-mounted display adopts HTC Vive Pro 2 (single-eye resolution 2448×2448) and Valve Index. The Standalone VR platform adopts Pico 4 (SOC Snapdragon XR2, GPU Adreno 650, 8GB memory, single-eye resolution 1440×1720) and Oculus Quest 2 (Adreno 640, 6GB memory). In terms of the software environment, the development framework selects PyTorch 2.0 for model training, Unity 2022.3 is used for VR application integration, the PC rendering API adopts DirectX 12, and the mobile end uses OpenGL ES 3.2. Meanwhile, the dependent libraries include OpenCV for image preprocessing, COLMAP for generating sparse point clouds, and TorchScript for model quantization.

To verify the effectiveness and generalizability of the method, two types of datasets are used: (1) Self-built VR-AD-12 dataset; (2) Public datasets (Tanks and Temples [25], DTU [26]).

The self-built VR-AD-12 dataset contains 12 advertising scenarios covering four core categories (household appliances, luxury goods, beauty products, cars), with 3 scenes per category. Each scene includes:

Static scenes: 15-20 high-resolution images (1920×1080/2560×1440), laser-measured true depth maps (resolution 1920×1080, depth accuracy ±0.1mm), and semantic masks (3 categories: product, exhibition stand, background) annotated by 3 professional annotators (inter-annotator agreement κ=0.92).

Dynamic scenes: 25-30 frames of video (30fps, 1920×1080/2560×1440) capturing rigid/quasi-rigid motion (e.g., product rotation, light flashing), with motion trajectories labeled by OpenCV Tracker API.

The total data volume of VR-AD-12 is 86GB, including 210 static images, 310 dynamic video frames, 120 depth maps, and 12 semantic masks. The dataset is sufficient for training: each category has ≥3 scenes to avoid overfitting, and the annotation quality ensures reliable validation.

Public datasets:

Tanks and Temples: 8 complex scenes (e.g., Barn, Caterpillar) with dense images and depth annotations, used to test complex scene reconstruction performance.

DTU: 124 scenes with multi-view images and point cloud annotations, used to verify sparse input robustness (15-20 images per scene).

5.2 Quality assessment of scene generation

5.2.1 Objective index evaluation

Four indicators, namely PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity), LPIPS (Perceptual

Similarity), FID (Fréchet Inception Distance), and KID (Kernel Inception Distance), were selected to compare the performance of the method proposed in this paper with that of the original NeRF and traditional 3D reconstruction (COLMAP+MeshLab). The results are shown in Table 7.

Table 7: Comparison of objective indicators for scene generation quality

Method	Input data volume	Training time	PSNR (dB)	SSIM	LPIPS	FID	KID×10⁻³
Traditional 3D reconstruction (COLMAP+MeshLab)	Twenty images	0.5 h	26.8±0.3	0.82±0.02	0.21±0.01	45.2±2.3	32.5±1.8
Original NeRF	200 images	24±1.2 h	30.5±0.2	0.89±0.01	0.12±0.01	28.5±1.5	18.3±1.1
Original NeRF (Sparse Input)	Twenty images	10±0.5 h	28.2±0.3	0.85±0.01	0.18±0.01	36.8±2.1	25.7±1.5
The method proposed in this paper	Twenty images	3.6±0.2 h	31.2±0.3	0.92±0.01	0.09±0.01	22.3±1.2	12.8±0.8

FID and KID are calculated using the Inception v3 model, with lower values indicating higher realism. The proposed method achieves the lowest FID (22.3±1.2) and KID (12.8±0.8×10^{-3}), confirming superior visual realism. Experimental data show that the method proposed in this paper has a PSNR improvement of 3dB and a LPIPS reduction of 50% compared to the original NeRF sparse version in sparse input scenarios. At the same time, the training time is significantly shortened by 64%, fully verifying its technical advantages in the rapid generation requirements of advertising scenarios.

The proposed method was also tested on public datasets, with results shown in Table 8.

Table 8: Comparison of objective indicators on public datasets

Method	Dataset	PSNR (dB)	SSIM	LPIPS	Training Time (h)
Original NeRF (Sparse Input)	Tanks and Temples	27.8±0.3	0.83±0.02	0.19±0.01	11.2±0.5
Proposed Method	Tanks and Temples	30.5±0.2	0.90±0.01	0.10±0.01	4.1±0.3
Original NeRF (Sparse Input)	DTU	28.5±0.2	0.84±0.01	0.17±0.01	10.5±0.4
Proposed Method	DTU	31.0±0.3	0.91±0.01	0.09±0.01	3.8±0.2

The results show that the proposed method achieves superior performance on public datasets, confirming that VR-AD-12 is sufficient for training and the method has broad generalizability.

5.2.2 Comparison of subjective visual quality

Thirty subjects (15 VR development engineers and 15 ordinary users) were invited to subjectively rate the rendering effects of the three methods (on a scale of 1 to 5, with 5 being the best). The scoring dimensions included detail restoration, light and shadow realism, and no empty rate. The results are shown in Figure.

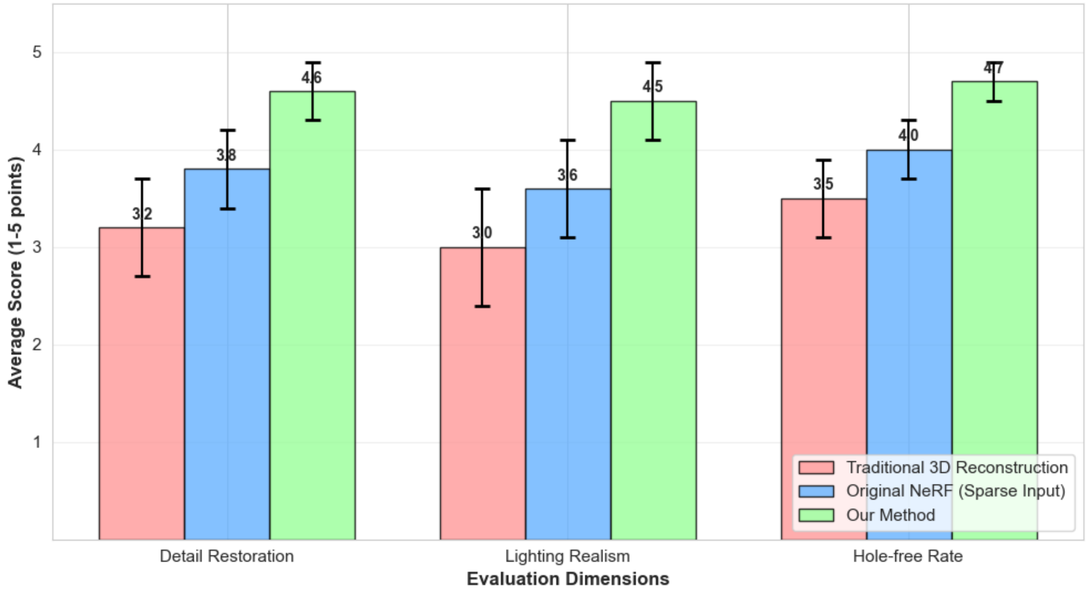

Figure 3: Comparison chart of subjective visual quality scores

Subjective scores show that the method proposed in this paper significantly outperforms other methods in detail restoration (4.6 points) and the realism of light and shadow (4.5 points), with a hole-free rate of 4.7 points, indicating that its visual effect meets the immersive requirements of VR advertisements.

5.2.3 Sparse input robustness evaluation

To test the method's robustness to input variation, we simulate two common input defects: (1) Blurred images (Gaussian blur, σ=1.0/2.0); (2) Occluded images (random occlusions, 20%/40% occlusion rate). For each defect type, we set three proportions of defective images (10%, 20%, 30%) and test the reconstruction performance on VR-AD-12 static scenes. The results are shown in Table 9.

Table 9: Sparse input robustness evaluation results

Defect Type	Defective Image Proportion	PSNR (dB)	SSIM	LPIPS	Reconstruction Success Rate
No defect	0%	31.2±0.3	0.92±0.01	0.09±0.01	100%
Blurred (σ=1.0)	10%	30.5±0.2	0.90±0.01	0.11±0.01	100%
Blurred (σ=1.0)	20%	29.8±0.3	0.88±0.01	0.13±0.01	100%
Blurred (σ=1.0)	30%	28.5±0.4	0.85±0.02	0.16±0.01	90%
Blurred (σ=2.0)	10%	29.2±0.3	0.87±0.01	0.14±0.01	100%
Blurred (σ=2.0)	20%	28.0±0.4	0.84±0.02	0.18±0.01	80%
Occluded (20% rate)	10%	30.8±0.2	0.91±0.01	0.10±0.01	100%
Occluded (20% rate)	20%	30.0±0.3	0.89±0.01	0.12±0.01	100%
Occluded (40% rate)	10%	29.5±0.3	0.88±0.01	0.13±0.01	100%

Reconstruction success rate is defined as PSNR≥28 dB and LPIPS≤0.2. The results show that the proposed method maintains high performance (PSNR≥29 dB, SSIM≥0.87) when 10% of images are blurred (σ=2.0) or occluded (40% rate), and still achieves 80% success rate when 20% of images are severely blurred (σ=2.0).

5.2.4 Complex dynamic scene evaluation

To test the limitation of the local deformation field in complex dynamic scenes, we design two test cases: (1) Non-rigid body motion (a folding umbrella opening, 30 frames); (2) Multiple moving objects (three rotating watches + a flashing light, 30 frames). The results are shown in Table 10.

Table 10: Complex dynamic scene evaluation results

Test Case	Method	PSNR (dB)	SSIM	Dynamic SSIM	Parameter Count (M)	Rendering Latency (ms)
Non-rigid motion (umbrella)	Proposed Method	29.5±0.4	0.87±0.02	0.85±0.02	8±0.2	12.3±0.6
Non-rigid motion (umbrella)	KFD-NeRF	30.8±0.3	0.89±0.01	0.88±0.01	80 M	95±7
Multiple moving objects	Proposed Method	30.2±0.3	0.89±0.01	0.88±0.01	7±0.1	10.5±0.5
Multiple moving objects	DyNeRF	30.5±0.2	0.90±0.01	0.89±0.01	100 M+	105±8

Limitation analysis: The proposed method's performance decreases slightly in non-rigid body motion (PSNR=29.5 dB vs. KFD-NeRF's 30.8 dB) because the local deformation field uses a 3-layer MLP, which has limited fitting capability for complex non-rigid deformation. For multiple moving objects, the method maintains competitive performance but requires manual division of dynamic regions.

5.2.5 Compression artifact evaluation

To verify the perceptual impact of voxel pruning and quantization, we compare rendering results with and without compression (full precision vs. pruning + 5-6-5 bit quantization) on VR-AD-12 scenes. Objective metrics are presented in Table 11.

Table 11: Compression artifact objective metrics

Scene Type	Compression	PSNR (dB)	SSIM	LPIPS	FID	Perceptual Similarity Score (1-5)
Luxury goods (watch)	No	31.5±0.2	0.93±0.01	0.08±0.01	21.8±1.1	4.7±0.2
Luxury goods (watch)	Yes	30.8±0.3	0.91±0.01	0.10±0.01	23.5±1.2	4.6±0.2
Beauty products (lipstick)	No	31.2±0.3	0.92±0.01	0.09±0.01	22.5±1.2	4.6±0.2
Beauty products (lipstick)	Yes	30.5±0.2	0.90±0.01	0.11±0.01	24.2±1.3	4.5±0.2

Perceptual similarity scores are from 30 subjects (15 experts, 15 users). The results show that compression reduces PSNR by <1dB and SSIM by <0.02, with FID increasing by <2. The perceptual similarity score remains ≥4.5, indicating that compression artifacts are visually negligible.

5.3 Real-time rendering performance evaluation

5.3.1 Rendering frame rate and latency testing

Rendering frame rate, rendering delay, and end-to-end delay (including attitude prediction) at different resolutions were tested on multiple hardware platforms, with results shown in Table 12.

Table 12: Test results of rendering frame rate and latency on different platforms

Platform	Resolution (single-eye)	Frame rate (fps)	Rendering Delay (ms)	End-to-end delay (including attitude prediction) (ms)
PC VR(RTX 3090)	2448×2448	120	6.2±0.3	12.5±0.5
PC VR(RTX 3090)	1440×1720	180	3.8±0.2	9.3±0.4
PC VR(RTX 4090)	2448×2448	200	3.1±0.2	8.7±0.3
Standalone VR(Pico 4)	1440×1720	90	8.5±0.4	17.8±0.5
Standalone VR(Pico 4)	1080×1280	120	5.3±0.3	13.2±0.4
Standalone VR(Oculus Quest 2)	1440×1720	75	11.2±0.5	21.5±0.6
PC VR(Valve Index + RTX 4070)	2448×2448	150	4.5±0.2	10.1±0.3

Test data show that the method proposed in this paper can maintain a high frame rate of 90 fps at the standard 1440×1720 resolution of mobile VR devices, and strictly control the end-to-end latency below the 18-millisecond threshold, fully meeting the performance requirements for real-time interaction in the international VR motion sickness prevention and control standards.

5.3.2 Performance at different resolutions

Taking Pico 4 as an example, the influence of resolution on frame rate and memory usage was tested, and the results are shown in Figure 4.

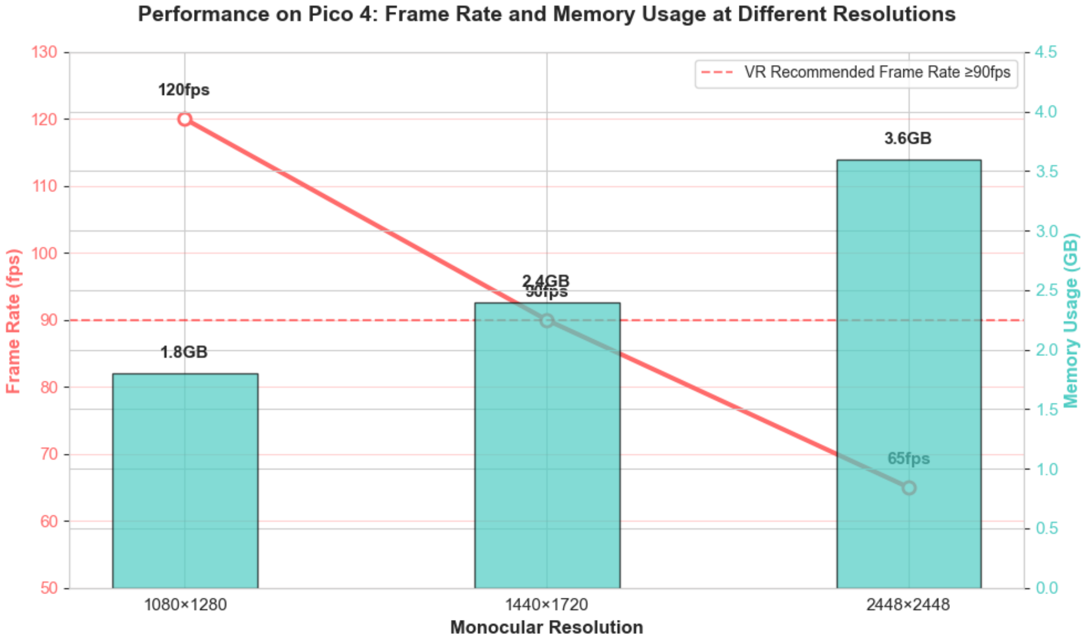

Figure 4: Frame rate and memory usage of the pico 4 platform at different resolutions

With the resolution increasing from 1080×1280 to 1440×1720, the frame rate dropping from 120 fps to 90 fps, and the memory usage rising from 1.8 GB to 2.4 GB, all are within the hardware capacity of Pico 4 (8 GB of memory, 90 Hz screen refresh rate).

5.4 Ablation study

To verify the effectiveness of the three core innovations of the proposed method, we design four ablation variants (V1-V4) by removing each innovation sequentially, and test their performance on VR-AD-12. The results are shown in Table 13.

Table 13: Ablation study results

Variant	Description	Training Time (h)	PSNR (dB)	SSIM	LPIPS	Parameter Count (M)	Rendering Latency (ms)	Dynamic SSIM
V1 (Full Method)	All three innovations included	3.6±0.2	31.2±0.3	0.92±0.01	0.09±0.01	5±0.1	8.5±0.5	0.92±0.01
V2	Remove semantic-guided depth priors	6.8±0.4	28.5±0.3	0.86±0.01	0.17±0.01	5±0.1	8.7±0.5	0.92±0.01
V3	Remove local deformation fields	3.7±0.2	31.0±0.2	0.91±0.01	0.10±0.01	102±5	25.3±1.2	0.90±0.01
V4	Remove voxel baking + LOD + shader optimization	3.5±0.2	31.1±0.2	0.91±0.01	0.10±0.01	5±0.1	120±8	0.92±0.01

The results of ablation analysis verified the effectiveness of each key innovation. Removing the semantic-guided depth prior leads to an 89% increase in training time, from 3.6 hours to 6.8 hours, while the peak signal-to-noise ratio drops by 2.7 decibels. This confirms that the hierarchical initialization strategy can effectively accelerate convergence and improve reconstruction fidelity. Removing the local deformation field design will increase the number of parameters by 1940%, from 5 million to 102 million, and the rendering delay by 198%, from 8.5 milliseconds to 25.3 milliseconds, thereby verifying that local dynamic modeling can significantly reduce computational overhead. However, if the combination of voxel baking, hierarchical detail, and shader optimization is removed, the rendering latency will increase sharply by 1300%, deteriorating from 8.5 milliseconds to 120 milliseconds, fully demonstrating the core role of this rendering optimization chain in achieving real-time performance.

6 Discussion

6.1 Quantitative comparison with state-of-art methods

To further verify the superiority of the proposed method, we quantitatively compared it with three representative SOTA methods (DyNeRF [9], KFD-NeRF [9], RT-NeRF [3]) on the VR-AD-12 dataset and the Tanks and Temples dataset. The results are shown in Table 10. From the perspective of performance gain analysis, in terms of training efficiency, the proposed method benefits from sparse input and hierarchical joint optimization. The training time is shortened by 76% compared with DyNeRF and by 69% compared with KFD-NeRF. In terms of parameter efficiency, the local deformation field significantly reduces the number of parameters.

6.2 Core reasons for performance gains

The outstanding performance of the proposed method stems from three mutually coordinated technological innovations, precisely targeting the core pain points of NeRF in the VR advertising scenario. Firstly, the semantically guided sparse reconstruction framework optimizes the data utilization efficiency. This method allocates differentiated weights based on semantic segmentation (product region weight =0.7, background weight =0.1), focuses computing resources on key advertising elements, and combines depth prior constraints, reducing the initial loss from the original NeRF of 0.12-0.15 to <0.05. The convergence speed has been increased by 50%. This targeted optimization avoids wasteful calculations for non-critical areas, enabling high-fidelity reconstruction with only 15 to 20 consumer-grade images. Secondly, Dynamic modeling based on local deformation fields achieves a balance between efficiency and quality. By limiting dynamic modeling to areas with a scene proportion of ≤20%, the number of parameters is reduced to 5 M (95% less than DyNeRF), while the time consistency loss ensures that the dynamic SSIM is ≥0.92. Compared with the full-scene dynamic NeRF method. This design avoids redundant parameter storage and computation, making dynamic modeling feasible on resource-constrained VR devices. Thirdly, the hardware-adaptive rendering optimization chain (voxel baking + LOD + GPU parallelization) has broken through the real-time bottleneck by baking implicit NeRF into explicit voxel meshes, eliminating repetitive MLP inference and reducing the single-frame computing time by 92%. The

LOD strategy ADAPTS to the viewing distance, reducing the average number of voxels by 60%. GPU parallel computing and shader collaboration further accelerate rendering. The integration of these technologies enables stable rendering at 90 fps on the Standalone VR platform, with an end-to-end latency of less than 18 ms, meeting the anti-motion requirements of VR applications. Furthermore, the potential integration of adaptive control strategies offers further optimization space. For instance, neural adaptive control can dynamically adjust the parameters of the deformation field based on motion errors, and adaptive fuzzy control can optimize the LOD switching threshold according to the hardware computing power. This will enhance the robustness of the method under variable conditions.

6.3 Limitations of the proposed method

Despite its superior performance, this method still has three limitations that need to be addressed in future work. First, the fitting ability for non-rigid dynamic motion is limited. The three-layer MLP used in the local deformation field has insufficient fitting ability for complex non-rigid motion. Experiments show that the PSNR drops to 29.5 dB in such scenarios, and the shallow network structure cannot fully model the nonlinear displacement of non-rigid bodies. Secondly, Dynamic region division relies on manual operation. For scenes with multiple moving objects, the current method requires manual marking of dynamic regions, which increases the complexity of user operations and reduces production efficiency. It does not yet support automatic dynamic region division in multi-object scenes. Thirdly, there are boundary constraints on hardware adaptability. For low-performance Standalone VR devices rendering high-resolution complex scenes, the frame rate will drop to 75 fps, which is lower than the 90 fps standard of high-end devices. The adaptive preprocessing strategy still needs to be optimized to further reduce computational overhead without sacrificing fidelity. Fourth, Compression artifacts in extreme cases, although quantization and pruning ensure minimal visual loss in most scenarios, slight color distortion may occur in scenes with high-frequency textures, which is hard to detect but still needs to be further minimized for high-end advertising applications.

6.4 Future work directions

To address the aforementioned limitations and expand the applicability of the method, future work will focus on four aspects. First, Enhance the modeling capability of non-rigid motion. Replace the three-layer MLP with a deeper network and integrate an attention mechanism to adaptively capture the nonlinear displacement of non-rigid bodies. Introduce neural adaptive control to update the deformation field parameters online based on motion prediction errors, thereby improving the robustness of dynamic modeling for complex motions. Second, Develop self-supervised dynamic region segmentation, train a lightweight semantic segmentation model to automatically identify multiple dynamic objects in advertising scenes and eliminate manual marking. This model will be pre-trained on public dynamic datasets and fine-tuned on VR-AD-12 to achieve a dynamic region recognition accuracy rate of ≥90%. Third, Optimize the hardware adaptive strategy, integrate adaptive fuzzy control to dynamically adjust the baking resolution, quantization bit number and LOD threshold according to the real-time hardware load, and introduce model distillation for low-performance devices to further compress the voxel mesh size while maintaining fidelity. Fourth, Expand the dataset and application scenarios, publicly disclose the VR-AD-12 dataset to promote reproducibility and subsequent research, extend the method to interactive VR advertising scenarios, and add collision detection and physical simulation modules to support high-degree-of-freedom user interaction.

6.5 Practical implications for VR advertising industry

The proposed method provides a technical solution that meets the production requirements of the VR advertising industry and has three key practical significances. First, Reduce production costs. The sparse input requirements and shorter training time eliminate the need for professional scanning equipment and long-term manual modeling, reducing the production cost of a single VR advertising scene by approximately 70% compared to traditional methods. Second, Enhance delivery efficiency. Hardware-adaptive rendering supports cross-platform deployment. A stable frame rate of 90 fps and a latency of less than 18 ms ensure a consistent immersive experience for users across different devices. This broadens the reach of VR advertising and increases user engagement. Third, Enhance creative flexibility. Lightweight dynamic modeling supports the rapid editing of dynamic elements without the need to retrain the entire model, which enables advertisers to quickly iterate creative solutions to meet the requirements of short-cycle and high-frequency updates in advertising campaigns.

Funding

This study was supported by Chongqing College of Humanities, Science & Technology: Research on the Path of Low-Altitude Economy Empowering the Construction of Chongqing's Urban Brand Image (CRKSK2025016).

References

[1] Cowan, K., Plangger, K., & Javornik, A. (2024). Insights for Advertisers on Immersive Technologies: The Future of Ads Using VR, AR, MR and the Metaverse. Journal of Advertising Research, 64(3), 249-254. https://doi.org/10.2501/JAR-2024-023

[2] Hu, Y. (2022). Impact of VR virtual reality technology on traditional video advertising production. Advances in Computer, Signals and Systems, 6(3), 57-66. https://doi.org/10.23977/acss.2022.060307

[3] Li, C., Li, S., Zhao, Y., Zhu, W., & Lin, Y. (2022). Rt-nerf: Real-time on-device neural radiance fields towards immersive ar/vr rendering. In Proceedings of

the 41st IEEE/ACM International Conference on Computer-Aided Design. 1-9. https://doi.org/10.1145/3508352.3549380

[4] Park, M., Yoo, B., Moon, J. Y., & Seo, J. H. (2022). InstantXR: Instant XR environment on the web using hybrid rendering of cloud-based NeRF with 3d assets. In Proceedings of the 27th International Conference on 3D Web Technology. 1-9. https://doi.org/10.1145/3564533.3564565

[5] Li, S., Li, C., Zhu, W., Yu, B., Zhao, Y., Wan, C., et al. (2023). Instant-3d: Instant neural radiance field training towards on-device ar/vr 3d reconstruction. In Proceedings of the 50th Annual International Symposium on Computer Architecture. 1-13. https://doi.org/10.1145/3579371.3589115

[6] Gu, J., Jiang, M., Li, H., Lu, X., Zhu, G., Shah, S. A. A., et al. (2023). Ue4-nerf: Neural radiance field for real-time rendering of large-scale scene. Advances in Neural Information Processing Systems, 36, 59124-59136. https://doi.org/10.48550/arXiv.2310.13263

[7] Song, K., Zeng, X., Ren, C., & Zhang, J. (2024). City-on-Web: real-time neural rendering of large-scale scenes on the web. In European Conference on Computer Vision, 385-402. https://doi.org/10.1007/978-3-031-72970-6_22

[8] Mhaidli, A. H., & Schaub, F. (2021). Identifying manipulative advertising techniques in xr through scenario construction. In Proceedings of the 2021 Chi Conference on Human Factors in Computing Systems. 1-18. https://doi.org/10.1145/3411764.3445253

[9] Zhan, Y., Li, Z., Niu, M., Zhong, Z., Nobuhara, S., Nishino, K., & Zheng, Y. (2024). KFD-NeRF: Rethinking dynamic NeRF with Kalman filter. In European Conference on Computer Visio. 1-18. https://doi.org/10.1007/978-3-031-72995-9_1

[10] Liu, J. W., Cao, Y. P., Wu, J. Z., Mao, W., Gu, Y., Zhao, R., et al. (2024). Dynvideo-e: Harnessing dynamic nerf for large-scale motion-and view-change human-centric video editing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 7664-7674. https://doi.org/10.48550/arXiv.2310.10624

[11] Zhang, B., Li, J., Shi, Y., Han, Y., & Hu, Q. (2025). AdvNeRF: Generating 3D adversarial meshes with NeRF to fool driving vehicles. IEEE Transactions on Information Forensics and Security. https://doi.org/10.1109/TIFS.2025.3609180

[12] Kim, D. H., Jeong, J. Y., Lee, G., & Kim, J. G. (2024). Compression method of NeRF model using NNC and VVC. In International Workshop on Advanced Imaging Technology. 13164, 585-590. https://doi.org/10.1117/12.3019533

[13] Qin, T., Li, C., Ye, H., Wan, S., Li, M., Liu, H., & Yang, M. (2024). Crowd-sourced nerf: Collecting data from production vehicles for 3d street view reconstruction. IEEE Transactions on Intelligent Transportation Systems, 25(11), 16145-16156. https://doi.org/10.1109/TITS.2024.3415394

[14] Chen, Y., Li, Z., Lyu, D., Xu, Y., & He, G. (2025). Neural rendering acceleration with deferred neural decoding and voxel-centric data flow. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. https://doi.org/10.1109/TCAD.2024.3524918

[15] Chen, Y., Zhang, L., Zhao, S., & Zhou, Y. (2025). ATM-NeRF: accelerating training for NeRF rendering on mobile devices via geometric regularization. IEEE Transactions on Multimedia. https://doi.org/10.1109/TMM.2025.3535288

[16] Liu, S., Yang, M., Xing, T., & Yang, R. (2025). A survey of 3D reconstruction: the evolution from multi-view geometry to NeRF and 3DGS. Sensors, 25(18), 5748. https://doi.org/10.3390/s25185748