Applied Intelligence (2025) 55:322
https://doi.org/10.1007/s10489-024-06005-9

STNeRF: symmetric triplane neural radiance fields for novel view synthesis from single-view vehicle images

Zhao Liu[1] · Zhongliang Fu[1] · Gang Li[1] · Jie Hu[1] · Yang Yang[1]

Accepted: 30 September 2024 / Published online: 14 January 2025

Abstract

This paper presents STNeRF, a method for synthesizing novel views of vehicles from single-view 2D images without the need for 3D ground truth data, such as point clouds, depth maps, CAD models, etc., as prior knowledge. A significant challenge in this task arises from the characteristics of CNNs and the utilization of local features can lead to a flattened representation of the synthesized image when training and validation with images from a single viewpoint. Many current methodologies tend to overlook local features and rely on global features throughout the entire reconstruction process, potentially resulting in the loss of fine-grained details in the synthesized image. To tackle this issue, we introduce Symmetric Triplane Neural Radiance Fields (STNeRF). STNeRF employs a triplane feature extractor with spatially aware convolution to extend 2D image features into 3D. This decouples the appearance component, which includes local features, and the shape component, which consists of global features, and utilizes them to construct a neural radiance field. These neural priors are then employed for rendering novel views. Furthermore, STNeRF leverages the symmetric properties of vehicles to liberate the appearance component from reliance on the original viewpoint and to align it with the symmetry of the target space, thereby enhancing the neural radiance field network's ability to represent the invisible regions. The qualitative and quantitative evaluations demonstrate that STNeRF outperforms existing solutions in terms of both geometry and appearance reconstruction. More supplementary materials and the implementation code are available for access at the following link: https://github.com/ll594282475/STNeRF.

Keywords Novel view synthesis · Single-view · Symmetric triplane · Rigid symmetry

1 Introduction

Novel view synthesis involves rendering 2D images of a 3D object or scene from new viewpoints, based on a set of 2D images and their known viewpoints. This technology is a crucial component of computer graphics and visual applications, particularly for virtual reality technologies such as Augmented Reality (AR), Mixed Reality (MR), and Virtual Reality (VR), MR, and VR, which have the potential to provide users with a more realistic and interactive immersive experience. Many studies [1–3] have utilized multiple views or integrated 3D geometric information to guide the understanding of geometry from images and generate 3D representations. However, obtaining multi-view images or 3D geometric data in real scenes is often challenging due to the requirement for substantial hardware and human resources. Thus, researching methods to obtain 2D images from new viewpoints using only a single image is a valuable direction, known as single-view-based novel view synthesis.

Additionally, a single image can only capture texture information from a fixed viewpoint, lacking comprehensive texture data and 3D structure. Recovering the spatial information of an entire complex scene from a single image is nearly impossible without the use of a real 3D model as a shape prior or the ability to leverage spatial consistency constraints between different views. To tackle this challenge, this

✉ Zhongliang Fu
fuzl@whu.edu.cn

Zhao Liu
rsliuzhao@whu.edu.cn

Gang Li
liggis@whu.edu.cn

Jie Hu
rsgis_hujie@whu.edu.cn

Yang Yang
2019102130012@whu.edu.cn

[1] School of Remote Sensing and Information Engineering, Wuhan University, Luo Yu Road, Wuhan 430079, Hubei, China

research focuses on the vehicles within the scene. Given that vehicles are frequently present in various types of images and are a common mode of transportation, inferring complete 3D information from a single vehicle image holds significant importance for applications such as autonomous driving. This work emphasizes the recovery of 3D structure information from single-vehicle images without any additional geometric information support.

While the NeRF [4] as well as subsequent Gaussian Splatting [5] have shown great potential in image-based novel view synthesis tasks and demonstrated a wide range of application prospects [6–8], most tasks still rely on multi-view images and are limited to fixed scenes, unable to share parameters between scenes. Some works [9–11] can recover 3D information from a single view but still require other views of the same target for validation during the training process. Collecting multi-view data of different targets in real scenes is challenging, and these methods still mainly remain in the stage of using synthetic datasets. When a single-view image simultaneously serves as both the network's input and validation, local features cause the network to collapse to a planar representation, making it unable to "imagine" information about unseen regions not included in the original image, as shown in the second row of Fig. 6. Therefore, some methods [12, 13] compress image features into one-dimensional global features to avoid this collapse. However, due to the limited expressive capacity of global features, the ability to describe details is constrained, as evidenced by the results of the AutoRF method in Fig. 4.

The core challenge of this paper is to address how to achieve the following using only a single vehicle image from a real-world scene, without relying on second or additional viewpoints: 1) Controlling network convergence; 2) Preserving the integrity of target rendering; 3) Fully utilizing local image information to enhance rendering quality. To tackle this challenge, the paper builds upon the baseline method AutoRF [13] and introduces Symmetric Triplane Neural Radiance Fields (STNeRF) for novel view synthesis based on a single vehicle image.

STNeRF employs a triplane representation, describing vehicle image features as three mutually orthogonal planes aligned with the target space, to extend 2D features to 3D. The triplane representation is renowned for its compactness and efficient expressiveness, with implicit properties that facilitate the learning of 3D structures through volumetric rendering [14]. However, in reconstruction tasks using a single view, triplane features are derived from 2D features based on the original viewpoint, limiting the amount of information available. In such cases, without additional 3D information or supervision from second or additional viewpoints during training, the features obtained during ray sampling can be erroneously guided to incorrect regions, as shown in the third row of Fig. 6. We first adopt a U-shaped network architecture to address this issue, designing the triplane feature extractor to capture more implicit information. Concurrently, we design spatially aware convolution to enhance communication between the three feature planes, thereby maintaining spatial consistency within the triplanes. Additionally, the rigid body symmetry of the vehicle is used to complement the 3D information and further constrain the spatial consistency of the triplane space, completing the reconstruction of the entire 3D vehicle information.

As demonstrated in Table 1, unlike prior methodologies, each ray within our constructed neural radiation field encompasses both global and local image features. Specifically, our STNeRF stands in contrast to latent vector methods, such as LolNeRF [12] and AutoRF [13], which compress structural and appearance features without individual latent vectors. This approach also diverges from methods that incorporate additional views or 3D models, such as PIFu [15], PixelNeRF [9], VITNeRF [10] and TripoSR [16]. In summary, our STNeRF approach seeks to improve the use of local features from a single image during training and validation, providing practical insights and techniques for future research in single-view-based neural radiation field reconstruction. Our main contributions are as follows:

- We introduce STNeRF, a NeRF-based rendering method that synthesizes novel views from real images with a single input.
- We propose a novel 3D representation that integrates global and local information using a triplane feature extractor with spatially aware convolution.
- We add an intrinsic rigid symmetry prior for vehicles to synthesize high-quality novel views with fine details regardless of pose transformation.
- We demonstrate state-of-the-art performance against existing approaches on real datasets.

The remainder of this paper is structured as follows: Section 2 provides a comprehensive review of the latest advancements in single-view novel view synthesis. Section 3 elaborates on the proposed STNeRF framework, detailing its network architecture and the underlying techniques. Section 4 presents experimental evaluations conducted on real-world datasets, including a comparative analysis with state-of-the-art methods. Finally, Section 5 concludes the paper by summarizing the key contributions and outlining potential directions for future research.

2 Related work

Recent advances in novel view synthesis from single-view images have garnered significant attention. This section provides a comprehensive review of the literature, organized into

Table 1 Comparisons with recent novel-view synthesis methods

	NeRF [4]	PIFu [15]	PixelNeRF [9]	VITNeRF [10]	TripoSR [16]	LolNeRF [12]	AutoRF [13]	STNeRF (Ours)
Learns scene prior	✓	✓	✓	✓	✓	✓	✓	✓
Supervision	2D	2D+3D	2D	2D	2D	2D	2D	2D
Image features	✗	✓	✓	✓	✓	✓	✓	✓
Viewer-centered coordinate	✗	✓	✓	✓	✓	✓	✓	✓
Global features	✗	✗	✓	✓	✓	✓	✓	✓
Local features	✗	✓	✓	✓	✓	✗	✗	✓
Single View only	✗	✓	✗	✗	✗	✓	✓	✓

three primary parts. Section 2.1 discusses methods for single-view reconstruction, including 3D mesh models, depth-based methods, signed distance functions (SDFs)-based methods, and implicit models. Section 2.2 reviews the NeRF methodology. Section 2.3 focuses on recent NeRF-based methods for generating novel views from single-view images.

2.1 Novel view synthesize for single-view images

3D mesh-based methods [1, 17–19] involve reasoning about mesh models from a single image. However, due to the limited expressiveness of low-resolution mesh models and the substantial computational resources required for high-resolution mesh models, these approaches often struggle to produce realistic results when applied to real-world datasets.

Depth-based methods [20, 21] typically employ depth estimation networks to perform one-sided 3D reconstruction, aiming for stereoscopic results. While these methods can yield high-quality 3D reconstructions in simple scenes, they face challenges in accurately inferring texture and structural information for missing regions, especially when dealing with scenes that lack complete information from a single viewpoint and involve incomplete depth estimations.

SDF-based methods [22] aim to address the limitations of depth-based approaches by representing geometric information using signed distance functions [23]. This representation helps maintain the smoothness of the reconstructed object's surface. However, it does not completely resolve the issue of the learning network's reliance on stereo information supervision.

Implicit model-based approaches represent an object or scene through a function, typically a neural network, that models the probability of a ray of light originating from a 3D viewpoint being obstructed by the object's surface. For instance, Park et al. [24] proposed DeepSDF networks, which implicitly encode the SDF for specific object classes. Genova et al. [25] introduced Local Depth Implicit Functions (LDIF), a 3D shape representation that decomposes space into a structured set of learned implicit functions. LDIF provides networks for inferring spatial decomposition and local depth implicit functions from 3D mesh data or pose depth images. [26] utilize coarse annotation information (object class and key points) on 2D images to learn category-specific 3D shape models from single images in a weakly-supervised manner, achieving state-of-the-art performance. However, they only focused on reconstructing the shape, without considering the reconstruction of appearance. Saito et al. [15] proposed Pixel Aligned Implicit Functions (PIFu), an implicit representation that locally aligns 2D image pixels with the global context of their corresponding 3D objects. PIFu utilizes a single photograph to reconstruct a human body's shape and color, employing implicit representation for mesh model reconstruction. However, it requires ground truth values of the human body model for supervision. Chibane et al. introduced IF-Net [27], which employed multiscale deep feature encoding to achieve state-of-the-art results in 3D geometric completion at the time. They further presented the approach [28] which combined local and global implicit representations with local texture information to infer global texture information. This work indicates that employing implicit representations like SDFs for 3D reconstruction may offer greater expressiveness compared to using point clouds or mesh models directly. Nevertheless, it still depends on the availability of 3D ground truth values.

2.2 Nerf

To address the dependency of novel view generation methods on 3D ground truth and the laborious processes of modeling, dynamic capture, and mesh generation in 3D rendering Mildenhall et al. [29] introduced an algorithm for view synthesis from an irregular mesh of sampled views. This method initially expands each sampled view into a local light field using a Multi-Plane-Image (MPI) scene representation and subsequently renders the novel view by merging adjacent local light fields. Following a similar trajectory, they subsequently developed NeRF [4], which advanced the novel

view synthesis task to a new level using implicit representation methods.

NeRF can render complex 3D scenes using only images with pose information as supervision. It generates rays from known views and samples points in 3D space along these rays. An MLP is then used to predict each sampled point's volume density and color. The colors and densities are weighted and accumulated using volume rendering to produce the final image. This approach enables NeRF to accurately reproduce complex scene details, significantly enhancing the quality and flexibility of novel view synthesis.

Despite NeRF's ability to produce high-quality novel views, it has significant limitations. Specifically, while NeRF can use implicit space features to "reverse render" images of novel views, it does not generalize well across different scenes, requiring retraining for each new scene. Furthermore, NeRF demands a substantial number of viewpoints for effective training. These limitations make learning 3D representations from a single view using NeRF a substantial challenge.

2.3 Nerf-based novel view synthesize for single-view images

Recent research has explored NeRF-based approaches for generating novel views from single-view images, with many methods demonstrating promising results. SinNeRF [30] by Xu et al. establishes a semi-supervised framework that enforces constraints on invisible regions, enabling their method to recover 3D information from complex scenes using a single image. However, it still relies on depth information as a prior constraint. Rematas et al. [31] proposed ShaRF, which constructs a geometric scaffold for an object and uses it to estimate the ray field. This is accomplished by reversing the process of estimating ray fields from a single image using both explicit and implicit representations. However, it still requires multiple views for constraints during the training phase. Chen et al. [32] introduced MVSNeRF, which constructs a cost volume by warping 2D image features to a reference view plane [33] as a means to reconstruct the neural codomain using a 3D convolutional neural network (3DCNN). The reconstructed data is then regressed through a Multilayer perceptron (MLP) structure similar to NeRF to render a novel view. MVSNeRF adopts a multi-view stereo reconstruction approach within the NeRF method, synthesizing a novel view from sparse inputs (3 views). This approach enhances the generalizability of the NeRF network but still relies on inputs from multiple views.

Kulhánek et al. [34] proposed Viewformer, which leverages the Transformer [35] and a vector quantization Variational Auto-Encoder (VQ-VAE) [36] to address the problem of sparse view inputs. However, it requires a substantial amount of data and effort to build the coding library. Yu et al. [9] introduced PixelNeRF, a method that integrates image features into the input of the NeRF network. This is accomplished by using a convolutional network to extract features from numerous images of a specific instance, allowing the network to gain prior knowledge of the instance's targets across various scenes. However, PixelNeRF still requires multiple views for training.

Müller et al. [13] presented AutoRF, which encodes each 3D target within a scene into a more compact representation, resulting in higher-quality reconstruction outcomes. However, AutoRF predominantly employs global features to enhance network generalizability, thereby generating novel views of somewhat limited quality. Rebain et al. [12] introduced Lolnerf, a method for reconstructing single-view face images using a single neural network that shares a latent space. Similar to AutoRF, Lolnerf predominantly relies on global features and overlooks local features.

Lin et al. [10] addressed the issue of missing local features by extracting global feature codes using Vit [37] and local feature codes using convolutional neural networks (CNNs). These feature codes are then fused and introduced as supplementary input to NeRF. However, this approach has limitations when applied in real scenes, as it uses original RGB values as a complement to light features and doesn't consider occluded information. Gu et al. [11] proposed NeRFDiff to address the challenge of generating detailed novel views based on monocular images without losing fine details. They refined the Conditional Diffusion Model [38] for 3D perception within NeRF, enabling the synthesis of finely detailed and 3D-consistent virtual views. However, this method relies on generative adversarial networks for learning and thus requires at least two views for training. Additionally, it incurs a relatively high computational cost during both learning and inference, given that the diffusion model is based on a Markov chain diffusion process. Tochilkin et al. proposed TripoSR [16], where images are encoded using DINO [39], and then a 64×64 resolution 3D triplane is generated through a transformer network. The triplane features are subsequently decoded into RGB colors, and new views are generated using the standard NeRF rendering method. Although TripoSR does not require viewpoint information of the images, it still needs to be trained on a multi-view image dataset.

The 3D Gaussian Splatting [5] method has recently garnered significant interest. By representing the scene as a collection of point clouds with Gaussian kernels and rendering directly in image space, this approach substantially improves rendering speed compared to NeRF methods. Szymanowicz et al. [40] proposed Splatter Image, which uses a 2D CNN to generate pseudo-images with colored 3D Gaussians per pixel, achieving efficient single-view 3D reconstruction with state-of-the-art performance, but requires multi-view supervision for training. Zou et al. [41]

introduced a hybrid Triplane-Gaussian representation, combining explicit and implicit representations to enable fast and high-quality single-view 3D reconstruction. However, generating a complete point cloud of the object from a single image is a challenging task that requires the 3D ground truth to supervise the generation of Gaussian kernels.

In summary, although substantial advancements have been achieved in single-view novel view synthesis, existing methods are constrained by several limitations. These limitations include the necessity for multiple views, extensive 3D training datasets, or significant computational resources. Differing from previous studies, this research introduces STNeRF, an end-to-end forward inference network model. STNeRF builds on the NeRF model and relies solely on single-view images for both training and inference, without requiring any prior 3D knowledge. Notably, STNeRF is designed specifically for vehicles, employing a dataset of single-view images to learn and generalize the implicit representation of vehicles. This allows for the reconstruction of a comprehensive 3D model of the vehicle from a single viewpoint

3 Methodology

In this section, we introduce the proposed STNeRF for single-view novel view synthesis, aiming to learn a general car neural rendering model from a single-view vehicle image without requiring any 3D ground truth data. We first introduce our Preliminaries in Section 3.1. Then, we provide a detailed description of the overall structure of STNeRF in Section 3.2. Finally, we introduce the loss functions used for training in Section 3.3.

3.1 Preliminaries

The preliminary preparation for STNeRF remains consistent with the baseline method AutoRF [13]. Here, we briefly review the preparatory work for creating system inputs, target space, and the generation and sampling of rays.

3.1.1 Inputs (I, M,res_{3d})

STNeRF initially preprocesses raw images containing multiple vehicles of interest using panoptic segmentation [42] to obtain cropped target images and instance segmentation results. For each vehicle target of interest, the cropped image is referred to as $I \in \mathbb{R}^{H\times W\times 3}$, and the segmentation result is denoted as $M \in \mathbb{R}^{H\times W}$, where h × w denotes the image resolution. The 3D object detection result $res_{3d} = [x, y, z, l, w, h, ry] \in \mathbb{R}^7$ in STNeRF is derived from the monocular 3D object detection method DD3D [43].

The variables x,y,z represent the 3D distance from the target's center to the camera viewpoint, while l, w, h represents the target's dimensions in 3D space, encompassing its length, width, and height. Furthermore, ry signifies the rotation angle of the target.

3.1.2 Normalized object coordinate space (NOCS)

The res_{3d} can represent a cuboid in the camera space that describes the pose and extent of the associated object. STNeRF employs the res_{3d} to establish the NOCS. Within the NOCS, the car's center is set as the origin, and the length, width, and height are scaled to a unit cube, i.e., $NOCS := \left[-\frac{1}{2}, \frac{1}{2}\right]^3$. The principal axis direction is aligned with the front of the car.

3.1.3 Ray generation and sampling

By generating the rays on the original image, the complete ray information of the scene is acquired, which is then projected onto the camera plane and cropped using the res3d to obtain the ray information of the target in the scene. A simple affine transformation can be used to convert the ray information from the coordinate system with the camera position as the origin to the NOCS. The rays generated within the NOCS are denoted as γ, and sampling points are obtained by sampling along the paths of these rays, represented as $\gamma_i = \gamma(t) = o + td$, where t the sampling distance.

3.2 STNeRF model

The architecture of STNeRF is depicted in Fig. 1. The inputs comprise an image depicting a detected 3D target and its corresponding instance segmentation mask. The target image undergoes processing through a feature extraction network designed to extract triplane features that encompass spatial information. These features include potential representations of both appearance and shape. Subsequently, these extracted features are input into separate appearance and shape decoders, which are implemented as MLPs to synthesize the color implicit network and the volume density implicit network, respectively. Finally, the novel view of the target is obtained using the volume rendering method proposed in [4].

Furthermore, the rigid symmetry of the vehicle is used to further constrain the spatial consistency within the triplane space.

Throughout the entire network, STNeRF only require the image of a single view of the target vehicle, without requiring any additional view information.

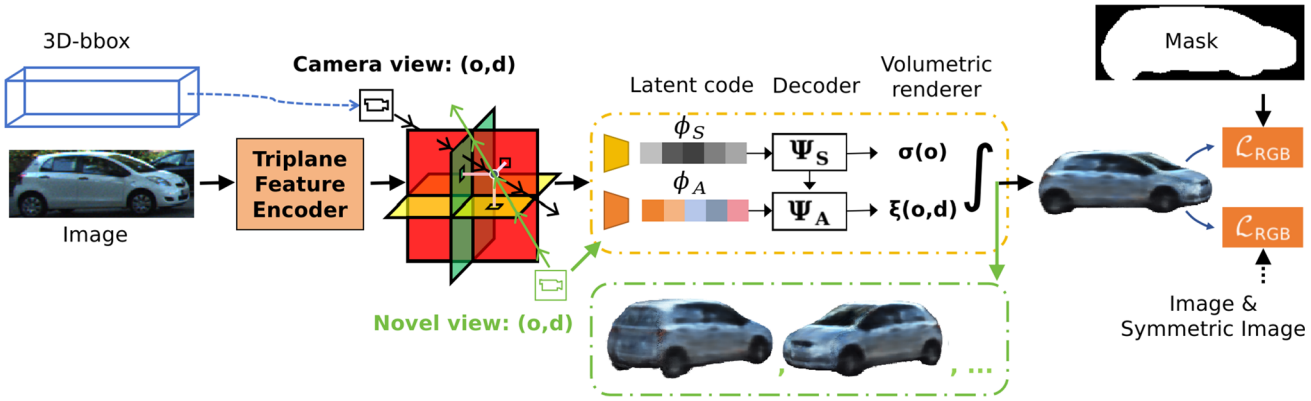

Fig. 1 STNeRF architecture. The Triplane feature Encoder is employed to encode the input images, yielding the separate latent representations of shape, ϕ_S, and appearance, ϕ_A. The latent representations ϕ_S and ϕ_A, along with ray information, are then input into the shape decoder Ψ_S and appearance decoder Ψ_A to synthesize the volume density implicit network σ and the color implicit network ξ, respectively. Finally, the novel view images are rendered using a volumetric rendering method

3.2.1 Triplane feature encoder

The vehicle novel view images 4 generated by AutoRF [13] show that while the local features of the vehicle are not perfect, the overall vehicle outline aligns with typical human visual expectations. This suggests that position encoding and global features contribute to a reasonable representation of vehicle shapes. Therefore, We design a triplane feature encoder using CNNs to encode shape and appearance latent codes from the target image I, denoted as $F_{Tri} = (\phi_S, \phi_A)$. In this context, ϕ_S represents a shape latent code encompassing global features, while ϕ_A constitutes an appearance latent code characterized by intricate details and maintaining spatial consistency. Specifically, a U-shaped network structure is employed to reconstruct the triplane representation of image features, with the three planes being mutually orthogonal, allowing for a partial mapping of 2D features to 3D space through learning. Additionally, a structure for maintaining spatial consistency between planes has been designed. The network structure is illustrated in Fig. 2.

The entire network structure is primarily divided into two main components: the encoder and the decoder. The encoder is designed with reference to the ResNet34 network architecture, comprising multiple convolutional and pooling layers to extract both low-level and high-level features from the image and project them into a high-dimensional representation. On the other hand, the decoder is composed of a sequence of convolutional and upsampling layers that gradually reduce the features extracted by the encoder to the original resolution. The connection between the encoder and decoder is facilitated by a series of skip connections. Their primary function is to transmit the low-level features learned by the encoder to the decoder, enhancing the construction of the triplane feature representation with more meaningful structural information.

The decoder's output features are straightforwardly divided into three distinct representations of uv, uw, and vw planar features, and these representations are orthogonalized to create a triplane feature $\phi_A = (3, u, v, w)$ that represents the appearance. The triplane features are assigned to each ray from sampling points in the target space through spatial sampling. They are then provided as input to the appearance decoder Ψ_A, which disentangles the color implicit network ξ.

Furthermore, the low-level and high-level features that have been extracted by the encoder undergo compression into a one-dimensional feature ϕ_S through a combination of upsampling and multiple convolutional layers. This one-dimensional feature is subsequently input to the shape decoder Ψ_S to carry out the prediction for the volume density implicit network σ.

3.2.2 Spatially aware convolution

Despite the triplane features being orthogonal in pairs, they still express 2D features based on the original viewpoint. This can lead to the spatial features generated by the CNN maintaining translation invariance, resulting in a more flattened representation of the rendered image in 3D rendering. More specifically, due to the lack of spatial correlation consideration in CNNs, it frequently occurs that the same spatial coordinate point conveys inconsistent information across the three planes, ultimately resulting in image distortion.

Consequently, for each ray, STNeRF applies a process akin to 3D convolution on the spatial sampling points to maintain spatial consistency, which is referred to as spatially aware convolution. More precisely, at each sampling point $ray_i = (u_i, v_i, w_i) \in \mathbb{R}^3$, spatial sampling involves extracting rows and columns corresponding to the point's coordinates in the triplane feature ϕ_A, resulting in six vectors denoted as

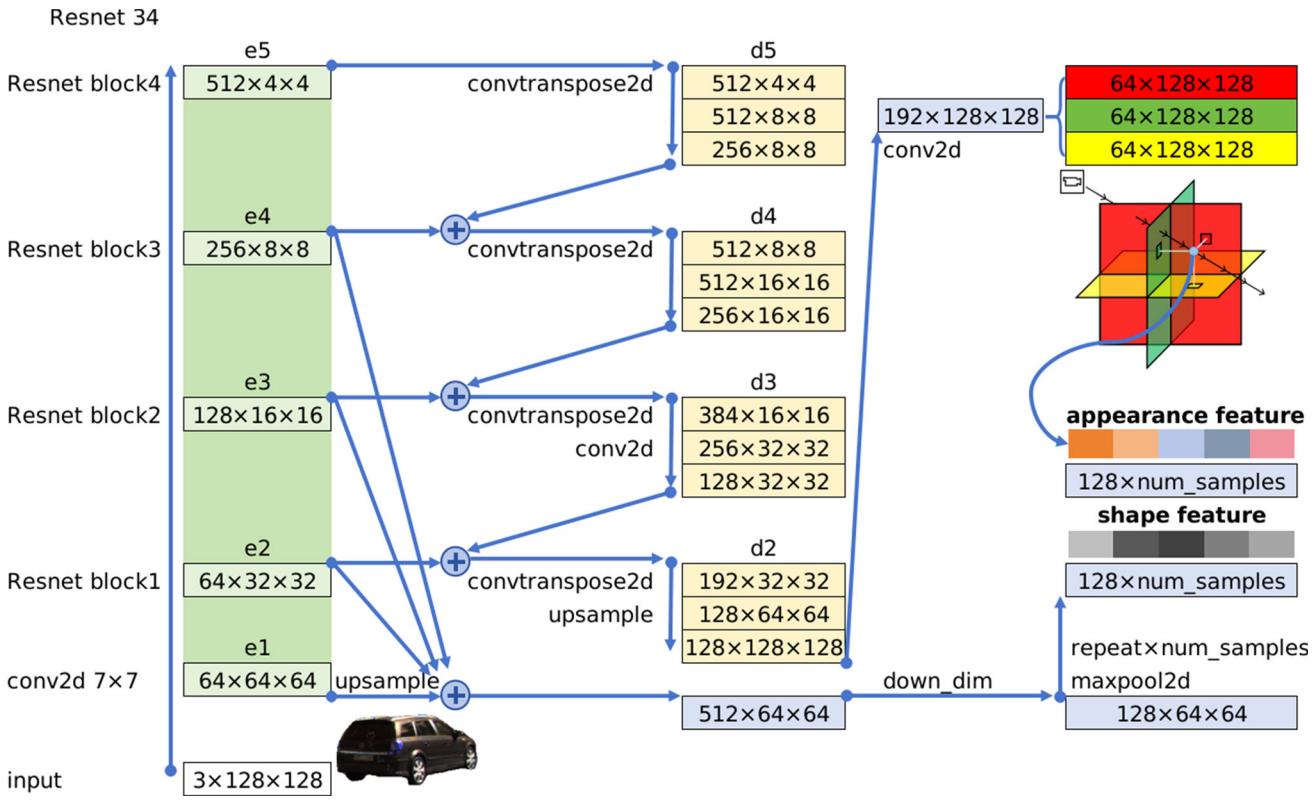

Fig. 2 Triplane feature encoder structure. Within the triplane feature encoder, input images are normalized to a resolution of 128x128x3, and their features are elevated to high dimensions using ResNet34. These high-dimensional features are subsequently upsampled and compressed to a single dimension, serving as a shape latent code. Concurrently, the high-dimensional features undergo deconvolution, convolution, skip connections, and upsampling to the normalized resolution, and are then decomposed into a triplane representation. This triplane representation, serving as the appearance feature, is then spatially sampled and assigned to each point along each ray

$uv_{i-u}, uv_{i-v}, uw_{i-w}, vw_{i-v}, vw_{i-w} \in \mathbb{R}^{1\times L\times Ch}$, where L represents the length of the rows and columns, and Ch signifies the number of feature channels. For the appearance feature $F(ray_i) \longleftarrow \phi_A$ of ray_i, it can be calculated by:

$$F(ray_i) = FC(FC(uw_{i-u} \oplus vw_{i-v}) \oplus FC(uv_{i-u} \oplus vw_{i-w}) \oplus FC(uv_{i-v} \oplus uw_{i-w})), \quad (1)$$

where FC denotes the fully connected layer and $\oplus$ is the concatenation function. Through this step, it becomes feasible to improve inter-plane communication, maintain spatial relationships without sacrificing local features, and achieve a spatially consistent information representation.

3.2.3 Shape decoder

The shape decoder Ψ_S takes the shape latent code ϕ_S and the coordinate position x of the sampling points as input, and outputs the implicit volume density network $\sigma \in \mathbb{R}^+$, denoted as $\sigma(x) = \Psi_S(\phi_S)$, for each sampling point.

3.2.4 Appearance decoder

The appearance decoder Ψ_A takes the appearance latent code ϕ_A, the implicit volume density network σ, the coordinate position x of the sampling points, and their corresponding ray direction d as input, and outputs the implicit color network ξ, denoted as $\xi(x, d) = \Psi_A(\Psi_S(\phi_S), \phi_A)$, for each sampling point. Ψ_A and Ψ_S are composed of MLP, with a similar network structure, but the difference is that it needs to decode the implicit volume density network σ, and it is related to the light direction. A detailed account of the specific network structure of Ψ_A and Ψ_S can be found in Section 4.2.

3.2.5 Rendering

The process of volume rendering follows the methodology employed in NeRF [4], with the distinction that we confine ray rendering exclusively to the target space. This process ultimately yields two essential outputs, denoted as $P(\gamma)$ and $C(\gamma)$. Here, $P(\gamma)$ represents the probability of the ray $\gamma(t) = o + td$ being intercepted within the target space along the d direction, while $C(\gamma)$ denotes the color accumulation

along this ray, specifically representing explicit pixel color information. The specific formula is defined as follows:

$$\begin{aligned} P(\gamma) &= \int_{t_n}^{t_f} T(t)\sigma(\gamma(t))dt) \\ C(\gamma) &= \int_{t_n}^{t_f} T(t)\sigma(\gamma(t))\xi(\gamma(t), d)dt)' \\ where T(t) &= exp\left(-\int_{t_n}^{t} \sigma(\gamma(s))ds\right), \end{aligned} \tag{2}$$

where t_n and t_f represent the nearest and farthest boundaries of the ray sampling, as determined by the target space generated through res_{3d}.

3.2.6 Rigid symmetry

STNeRF utilizes the left-right symmetry of the vehicle for two main purposes. Firstly, it addresses the reliance on global features in single-view reconstruction methods by incorporating local features from the triplane representation. However, the amount of local information contained in a single image is inherently limited. As a left-right symmetric rigid body, a vehicle offers sufficient symmetry, allowing for the effective supplementation of 3D features from a single image.

Secondly, constructing the triplane representation solely from a single image results in these planes always being orthogonal to the viewing direction. This limitation means that only the RGB values from known viewing angles can be used for validation during the training process. Consequently, when insufficient views are available to reconstruct the entire scene, the NeRF network tends to reduce complex scene information into a flat representation, as observed in [12].

We have provided a simple description of the local feature assignment during ray sampling in Fig. 3. When inferring the appearance information of the blue ray, directly mapping the sampling point (blue dot) to the view plane along the ray's direction might assign the features of the red point, which may be less relevant, to the visible region of the blue point. However, in reality, the green point carrying relevant features should be assigned to the blue point.

STNeRF is designed for vehicle scenes and assumes that for objects with intrinsic symmetries, the appearance of incoming light from the same spatial position on the left and right sides should be similar. It's important to note that since the triplanes are still synthesized from images orthogonal to the viewing angle, designing the triplanes as symmetric would still produce irrelevant features in camera space, especially in regions that are not symmetric to the original viewing angle. STNeRF employs a strategy in which, during spatial light ray sampling, half of the feature points of the sampled

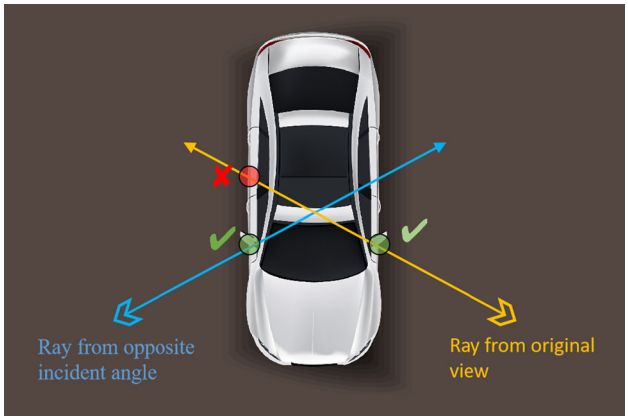

Fig. 3 Illustration of local feature assignment to sampling points of rays. The yellow lines depict the rays generated from the original viewpoint, while the blue lines represent the rays generated from the viewpoint that is symmetrical to the target space. The green dots indicate the features that should be assigned to the rays, whereas the red dots represent the features that are incorrectly assigned

light rays are randomly assigned to light rays on the opposite side with opposite incidence angles. This allocation serves to supplement the sampled rays during the training process.

Consequently, STNeRF naturally enforces symmetry constraints on the triplane features during their synthesis in the target space during training, enhancing the algorithm's robustness when generating views for the invisible region.

3.3 Training

Throughout the training process, rays are integrated to derive color information $C(\phi)$, which is subsequently regulated by the loss function L_{rgb}. While our method can produce the actual depth information $D(\phi) = o + P(\phi)d$, it's crucial to acknowledge that depth information in real-world images is frequently undisclosed. To address this, the method makes use of I_{mask} and $P(\phi)$ to formulate an occupancy loss function L_{occ} that incorporates predictions from the constrained implicit volume density network. In essence, the STNeRF relies on two distinct loss terms: an appearance loss function L_{rgb} and an occupancy loss function L_{occ}.

The appearance loss function L_{rgb} is defined as:

$$L_{rgb} = \frac{1}{n}\sum_{p\in n} \left\| I_p - C(\gamma_p \mid \sigma(x), \xi(x, d)) \right\|^2, \tag{3}$$

where n is the number of rays sampled, ϕ_p and I_p represent the sampled rays and their true color values.

To mitigate the presence of artifacts, STNeRF employs the Balanced Cross-Entropy Loss function to regulate $P(\gamma)$. The objective of the generated 3D model is to exhibit a distinct and perceivable appearance of a non-transparent medium.

Specifically, it aims for the total transparency resulting from the integration of rays to tend towards 0 or 1. The formula for the occupancy loss function L_{occ} is defined as follows:

$$L_{occ} = -[\beta P(\gamma) log I_mask\ (\gamma) + (1-\beta)(1-P(\gamma))log(1-I_{mask}\ (\gamma))], \quad (4)$$

where β represents the hyperparameter employed to balance the weighting between positive and negative samples, intending to minimize the occurrence of false negatives and false positives.

The ultimate loss function L is expressed as:

$$L = L_{rgb} + \lambda L_{occ} \quad (5)$$

where λ is a hyperparameter utilized to fine-tune the occupancy loss.

4 Experiments and analyses

4.1 Dataset

To illustrate the capacity of STNeRF in the generation of high-quality novel views from a single image in real-world conditions, the KITTI [44] and Nuscenes [45] datasets were selected to conduct arbitrary viewpoint image synthesis experiments based on single-view images.

The focus of STNeRF is the single-view images of vehicles within the datasets. The STNeRF method requires selecting vehicle images from the datasets based on two criteria: firstly, they should be within 50 meters from the camera; and secondly, they should have a minimum image resolution of 64x64 pixels, as determined by the instance segmentation results. As a result of this selection process, the KITTI and Nuscenes datasets yielded a total of 6255 and 30543 vehicle target images, respectively, accompanied by their corresponding 3D object detection and instance segmentation results, as illustrated in Fig. 1.

4.2 Implementation details

Our STNeRF was trained on 4 Nvidia Tesla V100 16GB GPUs, using a batch size of 4 and the Adam optimizer for 200 epochs. The learning rate was set to 10^{-4} for the first 100 epochs, 10^{-5} for epochs 100-150, and 10^{-6} for epochs 150-200. The hyperparameters β and λ were set to 0.75 and 0.5, respectively.

For the spatially consistent triplane encoder, the implementation details can be seen in Fig. 2. The encoder takes a single RGB image of size [3, 128, 128] as input and outputs appearance features and shape features of the same size $[num_p, 128]$. Here, num_p refers to the number of sampling points on multiple rays. In our method, the number of sampled rays is 2048, with 64 points uniformly sampled on each ray, so num_p is 2048×64.

The shape decoder is an MLP consisting of 5 ResNet blocks with a hidden dimension of 128. The input consists of the shape feature and the sampling point coordinates of size $[num_p, 3]$. The sampling point coordinates are encoded to higher dimensions through positional encoding [4], with a size of $[num_p, 60]$. The positional code of the sampling point coordinates is expanded to 128 dimensions through a simple linear layer to match the MLP hidden dimensions. It is then added to the shape latent code and passed through a ReLU activation function to introduce non-linearity. In each subsequent ResNet block, the positional code of the sampling point coordinates and the shape latent code are both input to the network and added to the feature map of the previous layer. Finally, a linear layer outputs the density value of size $[num_p, 1]$ for the sampling points.

The appearance encoder takes as input the appearance features and the output of the final ResNet block in the shape decoder. These two features are individually processed through a linear layer and then combined to create fused features. Furthermore, the direction of the rays at the sampling points, with a size of $[num_p,3]$, is encoded to $[num_p, 24]$ through direction encoding [4]. Subsequently, this encoded direction is expanded to 128 dimensions using a linear layer and added to the fused features obtained from the previous step. Finally, the features are passed through a ReLU activation function and a linear layer to produce the RGB values of size $[num_p, 3]$ for the sampling points.

4.3 Metrics

STNeRF employs several objective metrics to assess the disparity between the generated image and the original image within the initial viewpoint. These metrics include Peak Signal to Noise Ratio (PSNR) [46], Structural Similarity Index (SSIM) [46], Learning Perceptual Image Block Similarity Score (LPIPS) [47], and Frechet Distance Score (FID) [48].

PSNR, a widely recognized standard for quantifying image quality, serves to gauge the extent of distortion within the generated image. Higher PSNR values signify lower distortion. SSIM is a metric that evaluates the structural similarity between images, encompassing factors such as brightness, contrast, and structure. Larger SSIM values indicate greater similarity between images. LPIPS is an image similarity metric grounded in depth features, aligning more closely with human perceptual assessment. In this context, higher LPIPS values signify a higher degree of image similarity. Conversely, FID computes the distance between feature vectors of real and generated images. It quantifies the similarity between the two sets of images by examining the statistical aspects of computer vision features within the orig-

Table 2 Quantitative result of the novel view synthesis

Method	Nuscenes				KITTI			
	PSNR↑	SSIM↑	LPIPS↓	FID↓	PSNR↑	SSIM↑	LPIPS↓	FID↓
PixelNeRF	18.25	0.459	0.236	160.60	18.32	0.491	0.223	145.10
CodeNeRF	18.44	0.462	0.241	146.32	**19.01**	0.574	0.330	–
AuroRF	18.94	0.491	0.223	145.10	17.88	0.623	0.245	164.83
STNeRF (Ours)	**20.07**	**0.758**	**0.157**	**137.66**	18.01	**0.716**	**0.182**	**136.16**

inal images. Smaller FID values indicate a greater similarity between the two sets of image collections.

4.4 Baseline

We quantitatively and qualitatively compared STNeRF with the baseline method, AutoRF. In addition, we also compared it with state-of-the-art methods such as PixelNeRF [9], CodeNeRF [49], and TripoSR [16]. It is worth noting that PixelNeRF, CodeNeRF, and TripoSR are trained in a multi-view setup, whereas our method, like AutoRF, is trained using single observations from the same instance.

4.5 Quantitative evaluations and analysis

The spatially consistent triplane feature extractor designed by STNeRF can delve into more discriminative fine-grained local features. It complements global features, thus ensuring the richness of features while preserving volumetric density information. The experimental results presented in Table 2 demonstrate that our proposed STNeRF method consistently outperforms the benchmark AuroRF method across several image evaluation metrics. Specifically, on the Nuscenes dataset, STNeRF achieves an average increase of 5.9% in PSNR, an average enhancement of 54.2% in SSIM, an average reduction of 29.6% in LPIPS, and an average reduction of 5.4% in FID compared to AuroRF. On the KITTI dataset, STNeRF maintains its advantage with an average increase of 0.7% in PSNR, an average enhancement of 14.9% in SSIM, an average reduction of 25.7% in LPIPS, and an average reduction of 23.5% in FID over the AuroRF method.

The marginal improvement in the PSNR index with our algorithm can be attributed to its computation of mean square reconstruction error through least squares, a metric that leans toward favoring smoother or less textured images. Conversely, the SSIM index accounts for the reconstruction quality of image edges and high-frequency components, where our algorithm excels. Notably, our algorithm's performance is most evident in the LPIPS and FID indices, aligning more closely with human perceptual assessments. Overall, the test results unequivocally affirm the superiority of this algorithm.

4.6 Qualitative evaluations and analysis

Furthermore, given that STNeRF is designed for single-image-based novel view synthesis, the quantitative analysis presents challenges due to the absence of concrete numerical measures for evaluating the novel view. To provide a more comprehensive assessment of the novel view synthesis quality, we conduct a qualitative analysis of the novel view

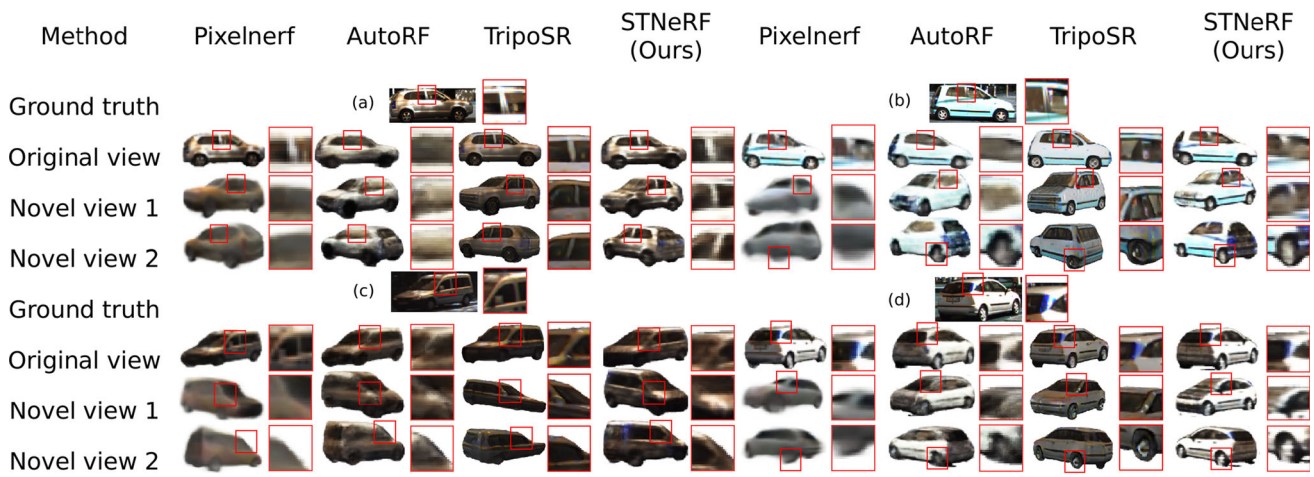

Fig. 4 Novel view synthesis results in target space

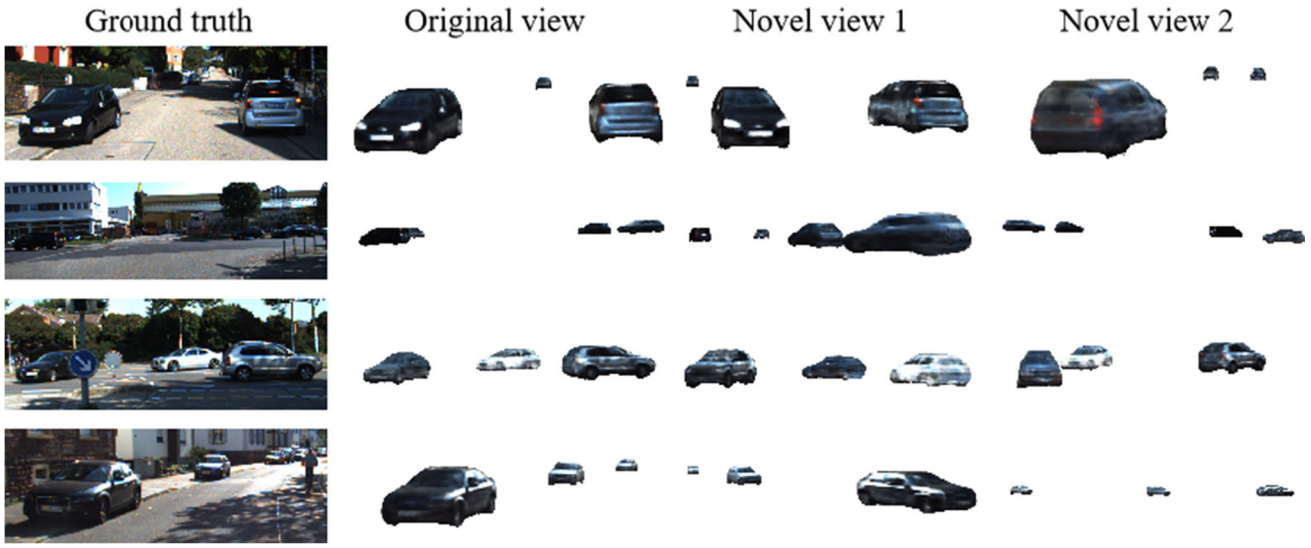

Fig. 5 Novel view synthesis results across the entire scene

synthesis results. The outcomes of the novel-view synthesis visualization on the Kitti dataset are depicted in Fig. 4.

From the results, it is evident that PixelNeRF excels in generating higher-quality views from the original viewpoint, primarily due to its comprehensive utilization of local features for inference. However, when moving away from the original viewpoint (novel views 1 and 2), PixelNeRF exhibits instability and blurriness. This phenomenon can be attributed to the fact that PixelNeRF acquires information for each sampling point via coordinate projection transformation to the original viewpoint, and the image features derived from 2D CNN lack the capacity to capture the intricate details of the entire 3D scene.

AutoRF, on the other hand, combines global features with positional coding, resulting in plausible results for generated novel views. Nevertheless, it struggles to control fine details due to the limited descriptive capabilities of global features.

TripoSR also employs a triplane reconstruction approach similar to STNeRF. However, TripoSR tends to lose lighting details, resulting in a generally darker overall appearance. Moreover, due to its design that does not require viewpoint information, TripoSR often produces inaccurate shape results, as evident in Figs. 4 (c) and (d).

In contrast, STNeRF introduces triplanes to characterize spatial features and incorporates rigid symmetry. This approach effectively retains detailed features and mitigates the flatness associated with 2D CNN to a certain extent. As observed in the results, such as the rendering of the shadow and the delineation of the black line beneath the vehicle in Fig. 4(b), as well as the representation of the wheel hub in Fig. 4(d), our algorithm preserves texture details while ensuring spatial consistency.

Additionally, we conducted experiments in Figure 5. to visualize the generation of novel view synthesis across the entire scene.

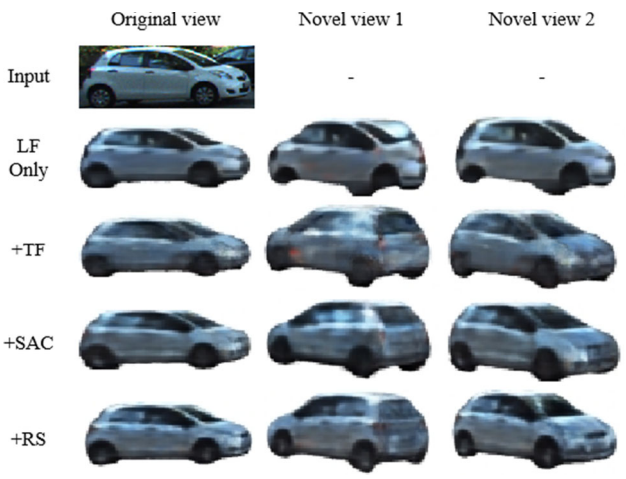

Fig. 6 Rendering performance under ablation settings. Here, LF refers to local features, TF refers to triplane features, SAC represents spatially aware convolution, and RS stands for rigid symmetry

Table 3 Quantitative analysis under ablation settings

	PSNR↑	SSIM↑	LPIPS↓	FID↓
LF Only	**18.28**	**0.770**	**0.149**	**99.98**
+TF	17.54	0.661	0.211	173.29
+SAC	16.82	0.679	0.200	169.22
+RS	**18.01**	**0.716**	**0.182**	**136.16**

Here, LF refers to local features, TF refers to triplane features, SAC represents spatially aware convolutions, and RS stands for rigid symmetry

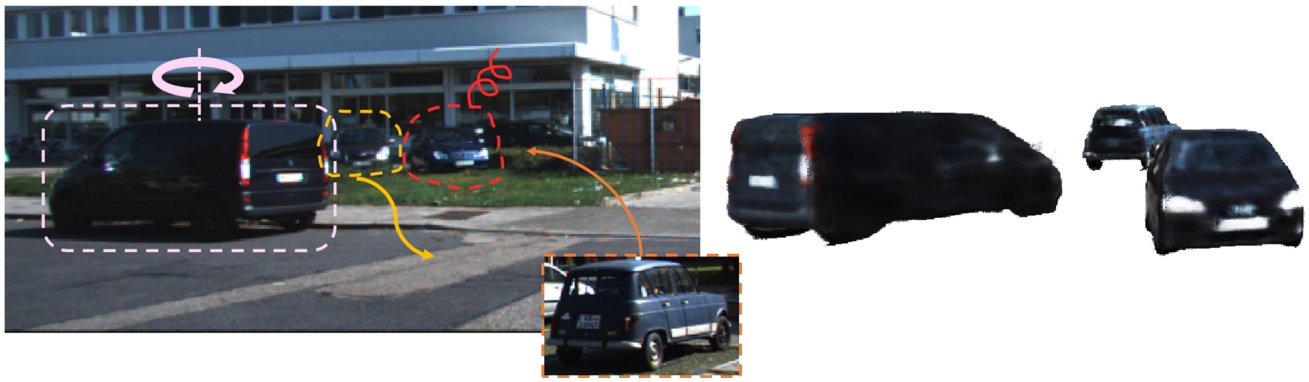

Fig. 7 Editing of the spatial position

4.7 Ablation studies

We conducted ablation studies to evaluate the effectiveness of our STNeRF method for novel view synthesis from a single image, as depicted in Fig. 6 and quantified in Table 3.

As a baseline, only single-view local features were used for reconstruction, with local features directly mapped to object-centered space and sampling point features obtained via bilinear interpolation. At this stage, the NeRF network rendered novel views flatly from the original viewpoint, producing satisfactory rendering results (row 3 of Fig. 6) and achieving the highest quantitative metric score (row 2 of Table 3). However, novel views from other viewpoints were entirely incorrect, demonstrating the limitation of only using local features. The high-quality results from the original viewpoint demonstrated only the most ideal performance of this method using local features.

When introducing the triplane feature extractor to extract local features and assign them to sampling points on rays, all metrics showed a decrease (row 3 of Table 3). This is attributed to our primary objective of generating delicate and realistic novel views. To encompass the comprehensive generation of spatial features, the features of sampling points on light rays in the original viewpoint inevitably cannot solely contain the local features of the original view. Furthermore, the network began to acquire spatial knowledge from 2D images (row 4 of Figure 6). The rendered car, when viewed from various angles, although somewhat blurry, already exhibited the appearance of a car. However, spatial inconsistencies were observed during novel view rendering, such as the rear part of the front wheel being erroneously rendered in red in novel view 1.

While adding spatially aware convolutions to the triplane features led to some improvements in SSIM, LPIPS, and FID scores, the PSNR score continued to decrease (Table 3, row 4), indicating that spatial consistency processing further down weighted local features. As shown in the fifth column of the rendered images in Fig. 6, although the images remain

Fig. 8 Fusion of the shapes and appearances. Here APP. means appearance

blurry overall, there were no obvious spatially inconsistent artifacts observed.

Finally, we modeled the vehicle as a rigid with symmetrical properties and incorporated this as prior knowledge into the training framework as the final model of STNeRF. As shown in the last row of Table 3, all evaluation metrics saw significant improvements. While the performance of using local features alone was not achieved for the original view, Fig. 6 demonstrates that rendering quality was markedly enhanced for novel views, with reasonable predictions made for regions not visible originally, such as vehicle details on the back (novel view 1) and front (novel view 2). Taken together, Table 3 and Fig. 6 indicate that our STNeRF method achieved an excellent balance of local features and global spatial information, leading to state-of-the-art results for the task of novel view synthesis from a single image.

4.8 Scene editing

STNeRF separates the rendering of pose, appearance, and shape information, enabling users to effortlessly manipulate the 3D scene. In scenarios involving a single instance or object, STNeRF not only facilitates the generation of images from various perspectives but also offers straightforward means to modify their shape or appearance. Figure 7 and Fig. 8 illustrate the results of scene editing using the STNeRF, where users can readily alter any attribute of a vehicle, including its position, appearance, or shape, and even add, remove, or duplicate elements. In Fig. 7, STNeRF demonstrates modifications to the spatial position of the target vehicle. This includes rotating the vehicle in the pink region, advancing the vehicle in the yellow region, removing the vehicle in the red region, and merging the vehicle in the orange region with a designated location. In Fig. 8, STNeRF showcases the fusion of the shapes and the appearances of two target vehicles, resulting in a vehicle with the appearance of the left vehicle.

5 Conclusion

This paper introduces Symmetric Triplane Neural Radiance Fields (STNeRF), a new method for novel view synthesis from a single image. STNeRF utilizes a spatially consistent triplane feature extractor with spatially aware convolution to extract local features from the input image and extends them into a 3D representation with three orthogonal plane features. This effectively captures fine-grained details while maintaining spatial consistency. By leveraging the intrinsic symmetry of targets, STNeRF can enhance features from the symmetric perspective of the single input view.

Experimental results demonstrate that STNeRF outperforms baselines that solely rely on local or global features, achieving a photo-realistic synthesis of new views while maintaining spatial coherence. Both quantitative and qualitative evaluations validate that STNeRF achieves state-of-the-art performance on the challenging task of single-view novel view synthesis.

Moreover, STNeRF allows for easy editing of the position, appearance, and shape of 3D scenes through manipulation of the extracted 3D information. It also provides functionality for observing targets or scenes from different viewpoints as well as modifying their shapes or appearances.

In our future research, we will consider the difficulties associated with existing single-view reconstruction methods, including the use of diffusion models to further improve the quality of synthetic images.

Author Contributions Zhao Liu: Formal analysis, Methodology, Software, Writing - original draft preparation. Zhongliang Fu: Conceptualization, Writing - review & editing. Gang Li: Software, Data curation, Writing - review & editing. Jie Hu: Writing - review & editing. Yang Yang: Writing - review & editing.

Data Availability All the datasets used in this manuscript are published and publicly available for research. References to data sources are provided in the manuscript.

Declarations

Ethical and informed consent for data used The research does not involve human participants and/or animals. Consent for data used has already been fully informed.

Competing Interests All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

References

1. Gkioxari G, Malik J, Johnson J (2019) Mesh r-cnn. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 9785–9795
2. Zakharov S, Ambrus RA, Guizilini VC, Park D, Kehl W, Durand F, Tenenbaum JB, Sitzmann V, Wu J, Gaidon A (2021) Single-shot scene reconstruction. In: 5th Annual conference on robot learning

3. Zakharov S, Kehl W, Bhargava A, Gaidon A (2020) Autolabeling 3d objects with differentiable rendering of sdf shape priors. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 12224–12233
4. Mildenhall B, Srinivasan PP, Tancik M, Barron JT, Ramamoorthi R, Ng R (2021) Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM 65(1):99–106 ISBN: 0001-0782 Publisher: ACM New York, NY, USA
5. Kerbl B, Kopanas G, Leimkühler T, Drettakis G (2023) 3d gaussian splatting for real-time radiance field rendering. ACM Trans Graph 42(4):139–1
6. Šlapak E, Pardo E, Dopiriak M, Maksymyuk T, Gazda J (2024) Neural radiance fields in the industrial and robotics domain: Applications, research opportunities and use cases. Robot Comput-Integr Manuf 90:102810. https://doi.org/10.1016/j.rcim.2024.102810
7. Li H, Zhang D, Dai Y, Liu N, Cheng L, Li J, Wang J, Han J (2024) Gp-nerf: Generalized perception nerf for context-aware 3d scene understanding. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 21708–21718
8. Li H, Gao Y, Zhang D, Wu C, Dai Y, Zhao C, Feng H, Ding E, Wang J, Han J (2024) Ggrt: Towards generalizable 3d gaussians without pose priors in real-time. arXiv preprint arXiv:2403.10147
9. Yu A, Ye V, Tancik M, Kanazawa A (2021) pixelnerf: Neural radiance fields from one or few images. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4578–4587
10. Lin K-E, Lin Y-C, Lai W-S, Lin T-Y, Shih Y-C, Ramamoorthi R (2023) Vision transformer for nerf-based view synthesis from a single input image. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision, pp 806–815
11. Gu J, Trevithick A, Lin K-E, Susskind JM, Theobalt C, Liu L, Ramamoorthi R (2023) Nerfdiff: Single-image view synthesis with nerf-guided distillation from 3d-aware diffusion. In: International conference on machine learning, PMLR, pp 11808–11826
12. Rebain D, Matthews M, Yi KM, Lagun D, Tagliasacchi A (2022) Lolnerf: Learn from one look. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1558–1567
13. Müller N, Simonelli A, Porzi L, Bulò SR, Nießner M, Kontschieder P (2022) Autorf: Learning 3d object radiance fields from single view observations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 3971–3980
14. Chan ER, Lin CZ, Chan MA, Nagano K, Pan B, De Mello S, Gallo O, Guibas LJ, Tremblay J, Khamis S et al (2022) Efficient geometry-aware 3d generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 16123–16133
15. Saito S, Huang Z, Natsume R, Morishima S, Kanazawa A, Li H (2019) Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 2304–2314
16. Tochilkin D, Pankratz D, Liu Z, Huang Z, Letts A, Li Y, Liang D, Laforte C, Jampani V, Cao Y-P (2024) Triposr: Fast 3d object reconstruction from a single image. arXiv preprint arXiv:2403.02151
17. Henderson P, Tsiminaki V, Lampert CH (2020) Leveraging 2d data to learn textured 3d mesh generation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 7498–7507
18. Liu S, Li T, Chen W, Li H (2019) Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 7708–7717
19. Goel S, Kanazawa A, Malik J (2020) Shape and viewpoint without keypoints. In: Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16, Springer, pp 88–104
20. Wu S, Rupprecht C, Vedaldi A (2020) Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1–10
21. Dahnert M, Hou J, Nießner M, Dai A (2021) Panoptic 3d scene reconstruction from a single rgb image. Adv Neural Inf Process Syst 34:8282–8293
22. Muller N, Wong Y-S, Mitra NJ, Dai A, Nießner M (2021) Seeing behind objects for 3D multi-object tracking in RGB-D sequences. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 6071–6080
23. Oleynikova H, Millane A, Taylor Z, Galceran E, Nieto J, Siegwart R (2016) Signed distance fields: A natural representation for both mapping and planning. In: RSS 2016 workshop: geometry and beyond-representations, physics, and scene understanding for robotics. University of Michigan
24. Park JJ, Florence P, Straub J, Newcombe R, Lovegrove S (2019) Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 165–174
25. Genova K, Cole F, Sud A, Sarna A, Funkhouser T (2020) Local deep implicit functions for 3d shape. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4857–4866
26. Han J, Yang Y, Zhang D, Huang D, Torre FDL (2019) Weakly-supervised learning of category-specific 3d object shapes. IEEE Transactions on pattern analysis and machine intelligence (99):1–1
27. Chibane J, Pons-Moll G (2020) Implicit feature networks for texture completion from partial 3d data. In: Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16, Springer, pp 717–725
28. Chibane J, Alldieck T, Pons-Moll G (2020) Implicit functions in feature space for 3d shape reconstruction and completion. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 6970–6981
29. Mildenhall B, Srinivasan PP, Ortiz-Cayon R, Kalantari NK, Ramamoorthi R, Ng R, Kar A (2019) Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. ACM Transactions on Graphics (TOG) 38(4):1–14. ISBN: 0730-0301 Publisher: ACM New York, NY, USA
30. Xu D, Jiang Y, Wang P, Fan Z, Shi H, Wang Z (2022) Sinnerf: Training neural radiance fields on complex scenes from a single image. In: European conference on computer vision, Springer, pp 736–753
31. Rematas K, Martin-Brualla R, Ferrari V (2021) Sharf: Shape-conditioned radiance fields from a single view. In: International conference on machine learning, PMLR, pp 8948–8958
32. Chen A, Xu Z, Zhao F, Zhang X, Xiang F, Yu J, Su H (2021) Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 14124–14133
33. Yao Y, Luo Z, Li S, Fang T, Quan L (2018) Mvsnet: Depth inference for unstructured multi-view stereo. In: Proceedings of the European conference on computer vision (ECCV), pp 767–783
34. Kulhánek J, Derner E, Sattler T, Babuška R (2022) Viewformer: Nerf-free neural rendering from few images using transformers. In: European conference on computer vision, Springer, pp 198–216
35. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser u, Polosukhin I (2017) Attention is all you need. Adv Neural Inform Process Syst 30
36. Razavi A, Oord A, Vinyals O (2019) Generating diverse high-fidelity images with vq-vae-2. Adv Neural Inform Process Syst 32

37. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S (2020) An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929
38. Dhariwal P, Nichol A (2021) Diffusion models beat gans on image synthesis. Adv Neural Inf Process Syst 34:8780–8794
39. Zhang H, Li F, Liu S, Zhang L, Su H, Zhu J, Ni L, Shum H-Y (2023) DINO: DETR with improved denoising anchor boxes for end-to-end object detection. In: The eleventh international conference on learning representations. https://openreview.net/forum?id=3mRwyG5one
40. Szymanowicz S, Rupprecht C, Vedaldi A (2024) Splatter image: Ultra-fast single-view 3d reconstruction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10208–10217
41. Zou Z-X, Yu Z, Guo Y-C, Li Y, Liang D, Cao Y-P, Zhang S-H (2024) Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10324–10335
42. Kirillov A, Wu Y, He K, Girshick R (2020) Pointrend: Image segmentation as rendering. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9799–9808
43. Park D, Ambrus R, Guizilini V, Li J, Gaidon A (2021) Is pseudo-lidar needed for monocular 3d object detection? In: Proceedings of the IEEE/CVF international conference on computer vision, pp 3142–3152
44. Geiger A, Lenz P, Urtasun R (2012) Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE Conference on computer vision and pattern recognition, IEEE, pp 3354–3361
45. Caesar H, Bankiti V, Lang AH, Vora S, Liong VE, Xu Q, Krishnan A, Pan Y, Baldan G, Beijbom O (2020) nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 11621–11631
46. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing 13(4):600–612. ISBN: 1057-7149 Publisher: IEEE
47. Zhang R, Isola P, Efros AA, Shechtman E, Wang O (2018) The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 586–595
48. Heusel M, Ramsauer H, Unterthiner T, Nessler B, Hochreiter S (2017) Gans trained by a two time-scale update rule converge to a local nash equilibrium. Adv Neural Inform Process Syst 30
49. Jang W, Agapito L (2021) Codenerf: Disentangled neural radiance fields for object categories. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 12949–12958

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).
Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.
These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.
We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.
While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.
These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.
To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.
Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.
If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com