SCIENCE CHINA
Information Sciences

• REVIEW •

January 2026, Vol. 69, Iss. 1, 111101:1–111101:64
https://doi.org/10.1007/s11432-024-4541-9

Artificial intelligence for virtual reality: a review

Lili WANG[1], Weiwei XU[2], Yebin LIU[3], Miao WANG[1], Beibei WANG[4], Xubo YANG[5], Lan XU[6], Zhangyao TAN[1], Runze FAN[1], Zijun WANG[1], Chi WANG[2], Hongwen ZHANG[8], Yijian WEN[8], Haozhong YANG[1], Jian WU[1*], Jiahui FAN[9], Hui WANG[5], Qixuan ZHANG[6], Guoping WANG[7], Yongtian WANG[10] & Qinping ZHAO[1]

[1]*State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China*
[2]*State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310058, China*
[3]*Department of Automation, Tsinghua University, Beijing 100084, China*
[4]*School of Intelligence Science and Technology, Nanjing University, Nanjing 215163, China*
[5]*School of Software, Shanghai Jiao Tong University, Shanghai 200240, China*
[6]*School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China*
[7]*School of Computer Science, Peking University, Beijing 100871, China*
[8]*School of Artificial Intelligence, Beijing Normal University, Beijing 100875, China*
[9]*School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China*
[10]*School of Optics and Photonics, Beijing Institute of Technology, Beijing 100081, China*

Received 24 September 2024/Revised 25 March 2025/Accepted 8 May 2025/Published online 15 September 2025

Abstract As hardware and information technology continually advance, virtual reality (VR) has permeated numerous sectors, with applications becoming increasingly sophisticated. The evolution of VR systems has expanded from the seminal 3I characteristics—immersion, interaction, and imagination—to encompass 6I, incorporating intelligentization, interconnection, and iteration. The intelligentization of VR technology, an inevitable progression, has garnered heightened interest, particularly fueled by the emergence of artificial intelligence (AI) models and techniques like neural radiance fields, 3D Gaussian Splatting, neural rendering, generative adversarial networks, diffusion models, and large language models, which significantly propel the development of VR's core and pivotal technologies. This survey offers a comprehensive assessment of these pivotal VR technologies, harnessing the latest AI advancements, aiming to provide fresh perspectives and assist new researchers in staying abreast of groundbreaking work. We commence by detailing the acquisition process of reviewed papers, outlining our taxonomy grounded in VR's core elements and pivotal technological trajectories, and statistically analyzing the works within. Subsequently, we delve into the application of AI models, methodologies, and techniques across six research avenues: advanced AI-generated content representation, content rendering, content generation, physical simulation, virtual characters, and interaction, discussing their achievements. Concludingly, we summarize our findings, highlight existing challenges, and suggest potential avenues for future research.

Keywords virtual reality, artificial intelligence generated content, 3D Gaussian, avatar, physical simulation, interaction

Citation Wang L L, Xu W W, Liu Y B, et al. Artificial intelligence for virtual reality: a review. Sci China Inf Sci, 2026, 69(1): 111101, https://doi.org/10.1007/s11432-024-4541-9

1 Introduction

As hardware computing power persists in growing, significant advancements in artificial intelligence (AI) technology have ushered in breakthroughs across all fields of computer science. Researchers have harnessed AI's learning, reasoning, and generative abilities in the realm of virtual reality (VR) and associated domains. In 2023, Zhao [1] noted that virtual reality has evolved from version 1.0 with 3I features (immersion, interaction, and imagination) to version 2.0 with 6I features, with the newly added 3I features being intelligentization, interconnection, and iteration. Intelligentization encompasses intelligence in the creation and rendering of virtual content, physical simulations, avatar modeling, and human-computer interactions. For example, the advent of artificial intelligence generated content (AIGC) has garnered widespread attention, seamlessly integrating natural language text, images, and 3D data to swiftly produce high-quality 3D content. This provides a broader array of options for constructing virtual characters/agents and realistic virtual scenes within VR environments. The implicit depiction of radiation

* Corresponding author (email: lanayawj@buaa.edu.cn)

fields and the revolutionary concept of neural rendering, epitomized by the neural radiance field (NeRF), harnesses the full potential of AI's learning prowess, ushering in fresh perspectives on enhancing the rendering prowess of VR hardware devices. Furthermore, data-driven deep learning models find their niche in VR human-computer interaction scenarios, encompassing human body tracking, motion sickness anticipation, and attention analysis, among others. These models provide invaluable assistance in refining the dynamics of human-computer interaction, thus enhancing the overall VR experience.

The realm of intelligence in VR research spans a diverse array of disciplines. Several existing reviews encapsulate the integration of AI and VR technologies across domains like medicine [2], education [3,4], and industry [5], among others. Additionally, there are reviews that narrowly concentrate on pivotal AI applications within VR, encompassing realms such as advanced rendering techniques [6–8], object/scene generation [9–11], physical simulation [12,13], virtual character development [14–19], and innovative interaction methodologies [20–23]. Hirzle et al. [24] conducted a meticulous literature survey at the intersection of AI and extended reality (XR), meticulously organizing and classifying the findings. However, their endeavor primarily emphasized the compilation and categorization of literature, abstaining from delving into the technical intricacies. Oliveira et al. [25], on the other hand, presented a comprehensive literature review specifically targeting VR solutions leveraged by AI methods. Instead of categorizing the literature, they offered a holistic perspective, exploring the AI techniques that are most conducive to VR, the industries adopting AI in VR-based applications, the advantages and constraints of this integration, the emerging trends and opportunities it fosters, and the collaborative potential of AI and VR in facilitating smart manufacturing and logistics. Nonetheless, there remains a notable gap in the literature, as there is a scarcity of review articles that delve into and consolidate the solutions or optimization strategies brought forth by AI technologies rooted in the core content and technological underpinnings of VR itself.

In this paper, we embarked on a meticulous selection and analysis of 485 articles, with a focus on the 451 publications spanning the years 2018 to 2024, to provide an up-to-date overview of the pivotal technologies at the heart of integrating AI methods into VR. Our literature corpus was meticulously curated through a multifaceted approach, encompassing keyword-driven searches on Google Scholar and ArXiv, targeted queries in esteemed databases like IEEE Xplore, Association for Computing Machinery Digital Library (ACM DL), and Springer, as well as valuable recommendations from seasoned experts across diverse VR disciplines. After a rigorous assessment of the quality and relevance of the retrieved literature, we meticulously organized and synthesized the findings, grouping them into distinct research directions within the realm of virtual reality. This structured approach allowed us to present a comprehensive and nuanced understanding of the current state of research, highlighting the key technologies and advancements that are driving the integration of AI methods into VR applications.

Indeed, the organization of the selected articles into six distinct research directions—advanced AI-generated content representation, content rendering, content generation, physical simulation, virtual character, and interaction—offers a clear and comprehensive framework for examining the latest technological advancements in VR, fueled by AI theories and methodologies. Each of these directions showcases groundbreaking achievements that underscore the transformative impact of AI on VR applications. By presenting a summary of the current state of VR research through an AI lens, our review paper aims to serve as a valuable resource for a diverse audience. Researchers, both established and those embarking on their VR journey will find insights into the latest trends and challenges in the field. Graduate and undergraduate students aspiring to pursue VR research will benefit from synthesizing cutting-edge technologies and their theoretical foundations. Engineers working on developing VR systems will gain practical insights into how AI can enhance their designs and capabilities. Furthermore, individuals from related fields, such as artificial intelligence, game design, or human-computer interaction, will also find our review informative and inspiring, fostering cross-disciplinary collaboration and innovation. Lastly, our outlook on future trends in VR research powered by AI offers a glimpse into the exciting possibilities that lie ahead, inspiring readers to stay abreast of the latest developments and contribute to shaping the future of this rapidly evolving field.

In summary, in this AI for VR review, we have made the following contributions. (1) We have compiled the latest research outcomes on integrating AI in VR, utilizing targeted keyword searches and expert recommendations to ensure comprehensiveness. (2) We have methodically organized and categorized the interdisciplinary research content, unraveling the intricate technical pathways from diverse yet interconnected hot topics. (3) We have distilled the essence of existing AI-driven VR research, offering a holistic overview and a forward-looking perspective on emerging research directions.

Table 1 Searching source details.

	Database	Target	Keywords
Source 1	Google Scholar, arXiv	Collecting papers with high keyword relevance and relatively high citation counts	(virtual reality OR scene generation OR avatar OR agent OR simulation OR interaction OR rendering OR mesh OR model) AND (artificial intelligence OR deep learning OR network OR neural)
Source 2	ACM Digital Library, IEEE Xplore, Springer Link, Frontiers, MIT Press	Collecting papers in high-quality journals and conferences: *ACM TOG*, *IEEE TVCG*, IEEE ISMAR, IEEE VR, SIGGRAPH, CHI, ACM MM, UIST, NeurIPS, CVPR, ICCV, ECCV, AAAI, ICML, ICLR	
Source 3	Expert Experience	Collecting important papers recommended by experts in the field	

2 Methodology

In this section, we describe the scope of the literature survey, the search methodology and the literature database, and our evaluation methodology.

2.1 Scope

In this review, we will provide an overview of recent research in the field of artificial intelligence and deep learning applied to VR, with a focus on new methods or optimization techniques, so we start from what is involved in VR systems and search for literature covering at least one of the elements of virtual scenarios, virtual character, and human-computer interactions, and we further classify them under more specific elements.

2.2 Search process

Due to the broad scope of the VR domain, we searched Google Scholar[1)] using keywords covering VR, avatar, agent, interaction, rendering, scene, and model. We then focused on whether search results included artificial intelligence technologies, using keywords such as AI, deep learning, network, and neural. We also kept some of the highly cited unofficial articles on arXiv[2)] in the search results. Moreover, we searched the digital libraries of the ACM DL[3)], IEEE Xplore[4)], Springer Link[5)], Frontiers[6)], and MIT Press[7)]. We selected these databases because they cover major conferences like IEEE ISMAR and IEEE VR, as well as journal articles like *IEEE TVCG*. Additionally, significant contributions in virtual scene rendering and generation research can be found in artificial intelligence and computer vision journals and conferences such as CVPR, ICCV, and NeurIPS. The team of authors of this paper is also researchers in different directions of VR, and each author recommends some of the most classic or latest research articles that are closely related to his/her research direction. Table 1 gives the source details of the articles we cite.

We focus on relevant papers published between 2018 and 2024. This timeframe represents the boom period of AI research, and also comprehensively covers the crossover development stage between AI and VR.

2.3 Relevancy and quality assessment

We used some rules to evaluate the relevance and quality of the searched papers to filter the references cited in this paper.

Relevance assessment. To ensure that the screened references are within the scope of this review, we need to assess the relevance of the collected literature by addressing the following questions and keep the articles of high relevance as the subject of review in this paper.

(1) Does the research in this article belong to a particular direction in the VR field?

(2) Does this article utilize new AI techniques to solve problems in the VR field?

During the evaluation process, we found two issues that need attention.

(1) Because in keyword search, different popularity and bias of keywords may reduce the accuracy of search results. For example, keywords such as “simulation” and “modeling” may retrieve results from other research areas.

1) https://scholar.google.com/.
2) https://arxiv.org/.
3) https://dl.acm.org/.
4) https://ieeexplore.ieee.org/.
5) https://link.springer.com/.
6) https://www.frontiersin.org/.
7) https://mitpress.mit.edu/.

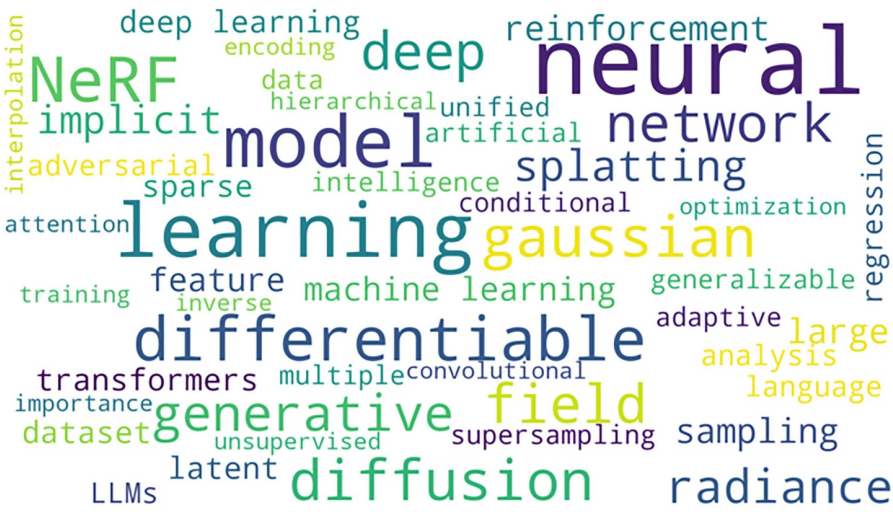

Figure 1 (Color online) Word cloud map of hot words in the titles of the reviewed papers.

We need to ensure that the search results are not overly biased in favor of AI techniques themselves, and we also need to exclude VR research based on traditional methods.

(2) Since we are using a combined search for virtual reality keywords and AI keywords, the search results also include articles that utilize virtual reality technology to generate training set data that can help AI train high-quality models. Given the goals of this paper, we also needed to exclude articles containing such content.

Quality assessment. After doing the relevance assessment, the quality assessment mainly focuses on finding high-quality and influential articles on new theories and methods of using AI methods to solve problems in VR. In the face of articles with similar research content, our selection priority is: expert-recommended articles from Source 3 are preferred; papers published in authoritative journals/conferences in the relevant fields from Source 2 are second because their academic rigor and authority have been verified; and lastly, articles found in Google Scholar or arXiv from Source 1 and with citation number greater than 10 are evaluated, because the content of these articles has been paid attention to by a larger number of researchers.

Finally, a total of 485 papers were obtained through the search, recommendation, relevancy, and quality assessment processes. We extracted the keywords in the titles of these papers, counted the word frequencies, and visualized the top 50 words as in Figure 1. The top ten words and occurrences are: neural 77, learning 47, model 43, NeRF 33, differentiable 31, Gaussian 27, diffusion 26, field 24, network 24, deep 23.

Items need to be clarified. (1) The counts of occurrences of NEURAL and FIELD exclude the NEURAL RADIANCE FIELD. (2) The counts of occurrences of LEARNING exclude the counts of MACHINE LEARNING and DEEP LEARNING. (3) The counts of occurrences of MODEL exclude the counts of the large language model. (4) The statistics of NeRF occurrences include NeRF and neural radiance field. (5) The statistics of DEEP occurrences exclude DEEP LEARNING. These high-frequency terms appearing in the titles of the papers indicate hotspots for research on AI models, methods, and techniques in VR.

3 Classification and statistics

Virtual scenes, virtual characters, and interactions are the most important core elements of virtual reality systems. These elements create a deep immersive experience that makes users feel like they are in a completely virtual environment. Therefore, we will further categorize the tasks of existing AI applications in virtual reality from these three perspectives: virtual scenes, virtual characters, and interactions. Figure 2 gives a taxonomy of tasks related to the core elements of virtual reality where AI algorithms are currently better applied. Due to the complexity of the virtual scene, we divided it into four sub-fields and summarized the hot AI-related directions in each sub-field.

Virtual scene. Virtual scene is the basis for creating an immersive virtual reality experience, usually a virtual world created through computer graphics technology, computer vision technology, etc. It can be a simulated real-life scene or a completely imaginary environment. Artificial intelligence applications in this domain include everything from scene generation to rendering. We classify existing virtual scene tasks into four categories: (1) advanced AI-generated content representation, (2) content rendering, (3) content generation, and (4) physical simulation. In

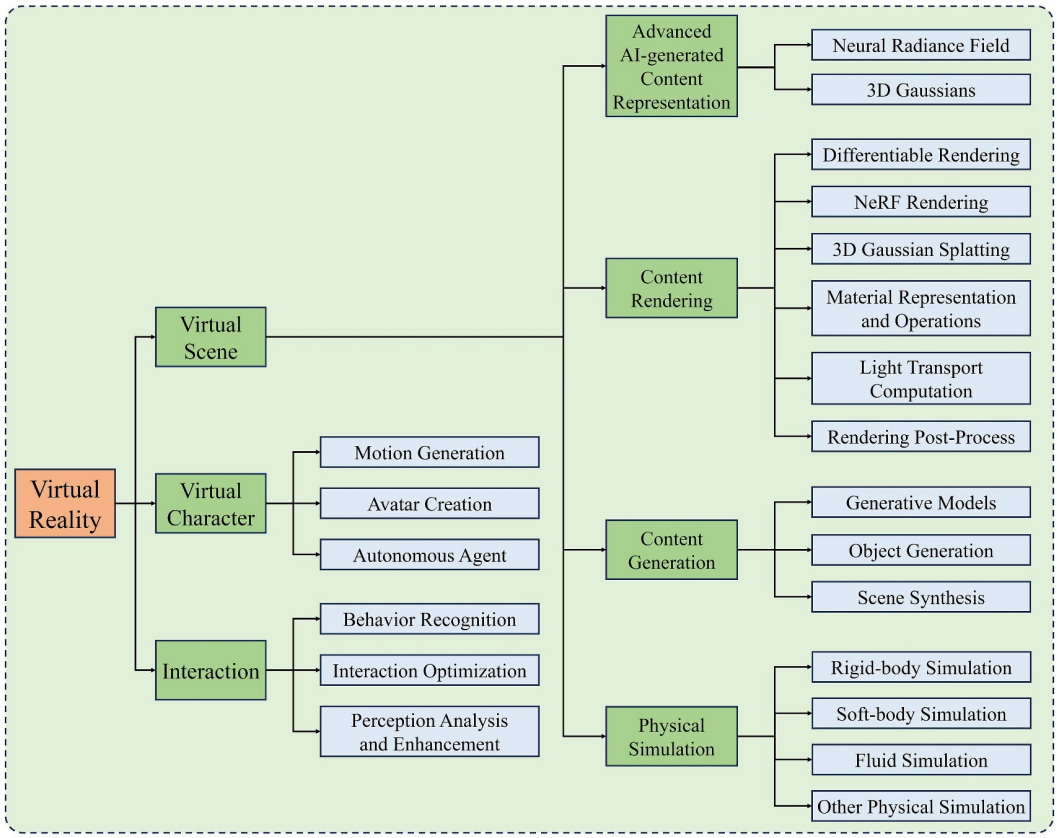

Figure 2 (Color online) Taxonomy of tasks.

scene representation, AI efficiently represents highly realistic scenes by constructing forms of complex environmental features that can be learned and simulated with neural networks. This representation not only enhances the scene's realism but also optimizes data storage and processing. In scene rendering algorithms, AI optimizes the entire rendering process by designing novel neural network architectures or data structures to accelerate the rendering process and improve visual quality. In scene generation algorithms, AI utilizes techniques such as generative adversarial networks and diffusion models to understand and reproduce real-world environmental features by training models. As a result, AI can generate realistic and detailed virtual environments and automatically create new scenes, thus reducing tedious and lengthy human overhead. AI creates more accurate and realistic physics effects in physics simulation algorithms by introducing new parameter estimation and pattern-learning methods to make virtual environments react more naturally.

Virtual character. AI plays a crucial role in the creation and behavioral simulation of virtual characters, enhancing their realism and interactivity. We classify the applications of AI in virtual character development into three categories: (1) motion generation, (2) avatar creation, and (3) autonomous agent. In motion generation, AI is used to interpret textual descriptions or to analyze audio inputs and generate corresponding movements for virtual characters, creating dynamic and contextually appropriate animations, enabling virtual characters to respond naturally to auditory stimuli and enhancing their interactivity. For avatar creation, AI-driven human reconstruction techniques focus on accurately mapping the geometric and texture details from real-world subjects, ensuring that the virtual representations are detailed and true to life. This precise modeling lays the groundwork for animatable avatars, which leverage AI to incorporate realistic animations and versatile movements, thus facilitating more engaging and personalized user interactions. In the area of autonomous agents, AI is used in crowd simulation algorithms to simulate the dynamic behaviors of large-scale virtual crowds, managing path planning and collision avoidance to ensure the naturalness and logic of group movements. Additionally, autonomous characters utilize AI behavioral inference models to predict and interpret actions, enabling them to make reasonable responses based on environmental changes and user inputs.

Interaction. Interaction is a pivotal element in user engagement with virtual content, functioning as the primary interface through which the virtual world is experienced. AI's impact permeates a wide spectrum of user interaction, from the recognition of user behaviors to the optimization of interaction techniques and the enhancement of overall

Table 2 Paper citations in sub-sections.

Section	Category	Refs.
	REVIEW	[1–29]
4.1	Neural radiance field	[30–34]
4.2	3D Gaussian	[35–39]
5.1	Differentiable rendering	[40–61]
5.2	NeRF rendering	[62–95]
5.3	3D Gaussian Splatting	[96–111]
5.4	Material representation and operations	[112–132]
5.5	Light transport computation	[133–156]
5.6	Rendering post-process	[157–181]
6.1	Generative models	[182–191]
6.2	Object generation	[192–242]
6.3	Scene synthesis	[243–257]
7.1	Rigid-body simulation	[258–263]
7.2	Soft-body simulation	[264–281]
7.3	Fluid simulation	[282–306]
7.4	Other physical simulation	[307–316]
8.1	Motion generation	[317–349]
8.2	Avatar creation	[350–402]
8.3	Autonomous agent	[403–422]
9.1	Behavior recognition	[423–456]
9.2	Interaction optimization	[457–473]
9.3	Perception analysis and enhancement	[474–485]

perception within virtual environments. We explore AI's contributions across three critical areas: (1) behavior recognition, (2) interaction optimization, and (3) perception analysis and enhancement. In behavior recognition, AI utilizes advanced neural networks, such as convolutional neural networks (CNNs), to precisely estimate and track user movements, including those of the hands, eyes, and facial expressions. This capability allows the virtual system to convert physical actions into meaningful digital inputs, fostering a more natural and responsive interaction with the virtual environment. Additionally, machine learning algorithms are deployed to deduce users' intentions based on their physical movements, thereby improving the accuracy and reliability of interaction inputs. In interaction optimization, AI mainly employs reinforcement learning and neural networks to enhance interaction methods within the constraints of physical space and hardware limitations. These techniques are meticulously designed to augment the intelligence and immersion of user interactions, broadening the possibilities within virtual environments and making user interactions more seamless and intuitive. In perception analysis and enhancement, AI utilizes a diverse array of contextual data, including sensor inputs and user behaviors, to model users' emotions and experiences. This process is primarily driven by machine learning regression models such as support vector machines (SVM) and random forests (RF). Through this analysis, AI enables the evaluation and optimization of virtual content quality, resulting in more personalized and enriched experiences that are finely tuned to the unique needs and preferences of individual users.

Based on the above taxonomy, we categorized our collection of 485 papers in Table 2 [1–485] to facilitate readers' quick access to papers relevant to the task of interest. Each category in the table corresponds to the secondary heading of our review. We also give a histogram of the distribution of the papers selected in the chapters corresponding to each secondary heading, guided by the above taxonomy (Figure 3). From this histogram, it can be seen that the number of papers we reviewed for section object generation and section avatar creation is greater than or equal to 50 papers, much more than the number of papers in other sections. This is because these two sections are now hot topics in VR, graphics, and 3D CV research, with numerous technological advances. In section neural radiance field and section 3D Gaussian, we only discuss the papers that first proposed these two representations and the paper that derived the principle of 3D Gaussian projection, and we put the other papers that modified these two representations to improve the quality or speed of rendering, as well as generalizing the scene, into the later sections NeRF rendering and 3D Gaussian Splatting to discuss in detail.

We also grouped the reviewed papers according to when they were accepted or published and gave a histogram of the distribution of the number of articles under different years (Figure 4). It is not difficult to see that the research on the application of AI technology in the field of VR is heating up year by year. Meanwhile, we are more inclined to review the latest papers, with the papers after 2020 reaching nearly 70% of the total number of papers reviewed.

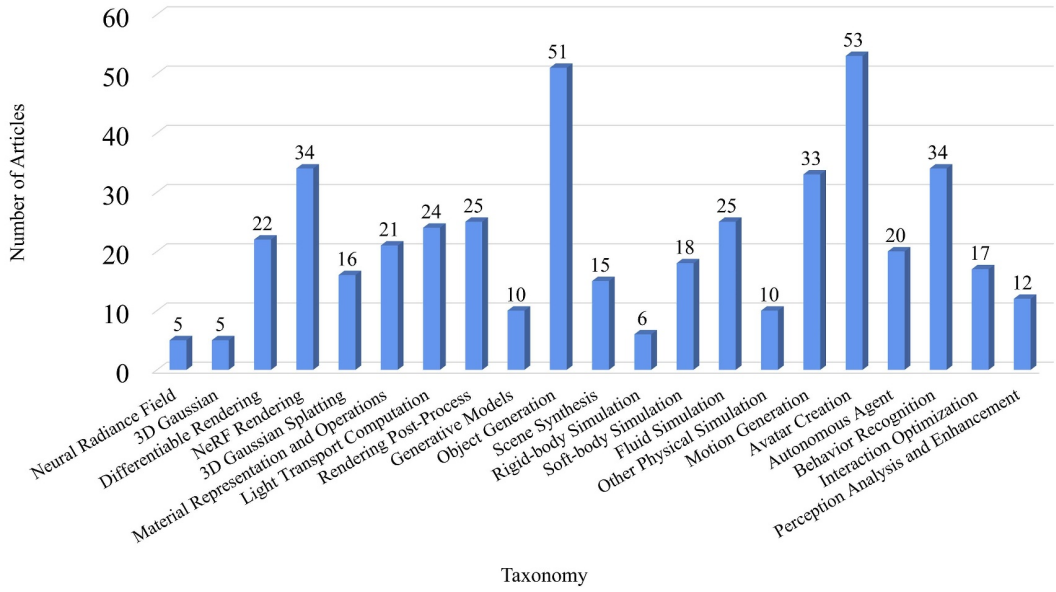

Figure 3 (Color online) Number of articles per task.

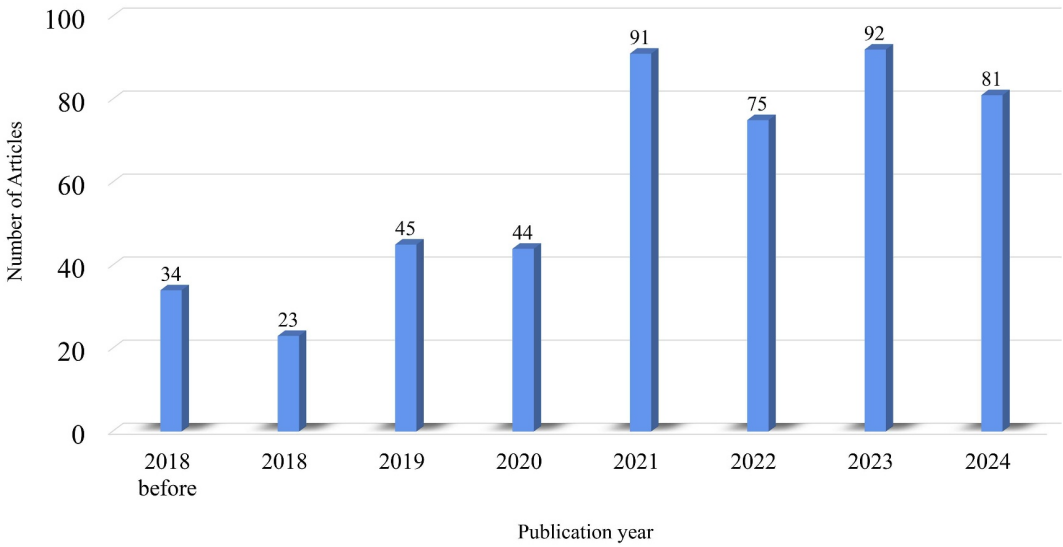

Figure 4 (Color online) Number of articles per year.

We grouped these papers according to the international journals and conferences in which they were submitted or published. Figure 5 gives a histogram of the distribution of the number of journals or conferences to which the papers we reviewed belong.

As can be seen from Figure 5, the papers we selected and reviewed are of high quality, with a large number of papers originating from *ACM TOG*, the top journal in graphics, and CVPR, the top conference in computer vision. Since the readers of this paper may have different research backgrounds, we also grouped the involved journals and conferences according to their fields, which include virtual reality, human-computer interaction, computer graphics, computer vision, and artificial intelligence.

The subsequent sections of this paper are organized as follows. Section 4 presents an overview of advanced AI-generated content representation. Section 5 reviews research related to content rendering, while Section 6 examines content generation approaches. Section 7 focuses on physical simulation techniques. Section 8 discusses virtual characters, and Section 9 explores interaction mechanisms. In Section 10, we summarize the current state of research in each area, outline key challenges, and propose potential directions for future work. Finally, Section 11 provides the concluding remarks.

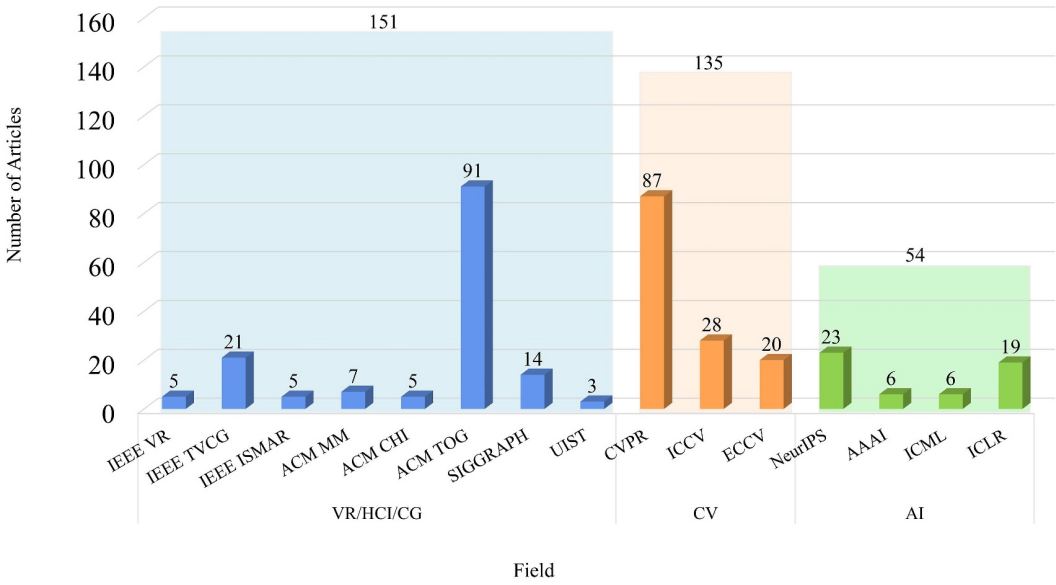

Figure 5 (Color online) Number of articles per field.

4 Advanced AI-generated content representation

In recent years, NeRF and 3D Gaussians, as advanced 3D content representation specifically designed for AI algorithms, have fully utilized the ability of deep learning models to process complex data and learn complex tasks by virtue of good end-to-end trainability, high flexibility, and adaptability, fine spatial representation, and gradient conductivity, and have been used in the fields of virtual reality, graphics, and computer vision attracted wide attention and applications. In this section, we will focus on reviewing and discussing these two advanced AI-generated content representations.

4.1 Neural radiance field

Neural radiance field was first proposed by Mildenhall et al. [30] in 2020. The core concept of NeRF is to represent a scene as a function of 3D locations and viewing directions by combining volume rendering techniques with typically implicit neural representations using multi-layer perceptrons (MLP), termed as a radiance field. This implicit radiance field representation optimizes network parameters using a set of images with known camera poses, enabling the learning of both geometric and photometric properties of 3D environments. For a 3D location (x, y, z) and a view direction (θ, ϕ), NeRF transforms the 5D vector with an MLP network $F_\Theta : (x, y, z, \theta, \phi) \rightarrow (\sigma, c)$, to get its corresponding volumetric density and view-dependent color. Then, employs a classic volume rendering algorithm to synthesize images from arbitrary new views. Figure 6 shows the overview of NeRF representation.

4.1.1 *NeRF construction*

Given a set of RGB images and the corresponding 6 degrees of freedom camera calibration parameters as inputs, NeRF outputs a representation of the geometry and appearance of a 3D scene. The geometric information is represented as a one-dimensional density value σ, and the appearance information is represented as multidimensional color features c. The color features c are then combined with the view direction d input into another MLP to obtain the view-dependent color feature vector. All these properties can be learned and optimized through gradient backpropagation. NeRF representation trains a network to map the 3D Cartesian coordinate into scene geometry and appearance. This neural network-based representation maintains multi-view consistency by decoupling the prediction of volume density σ from the view direction (θ, ϕ), while allowing color c to depend on both the view direction (θ, ϕ) and the 3D position (x, y, z). Specifically, this is achieved by designing two MLPs: the first MLP takes (x, y, z) only as input and outputs density σ and a high-dimensional feature vector, which indicates the view-independent color feature vector. This feature vector is then concatenated with the view direction (θ, ϕ) and passed to the second MLP, which outputs the view-dependent color c.

In traditional NeRF algorithms, the sampling process is divided into two stages: the coarse stage and the fine stage. During the coarse stage, the importance weight w_i of each sampling point is used as a piecewise constant

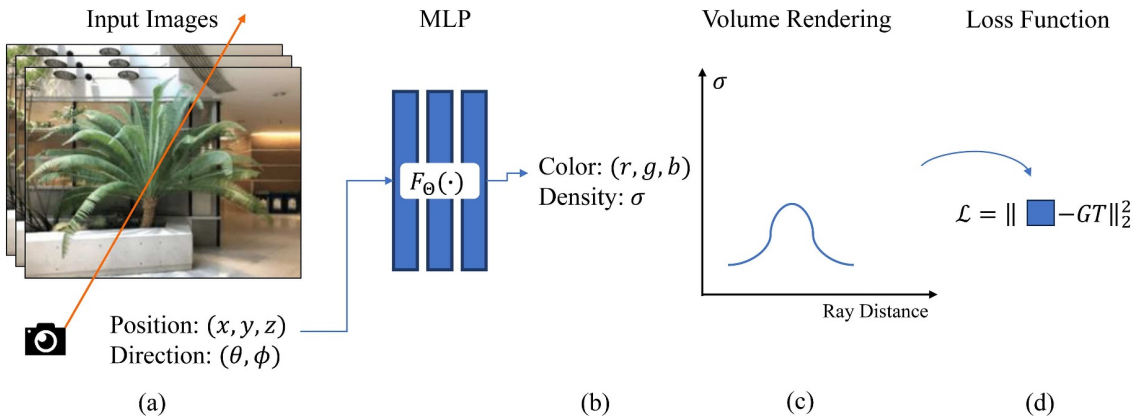

Figure 6 (Color online) An overview of NeRF scene representation and differentiable rendering procedure in the work of Mildenhall et al. [30]. (a) Sampling 5D coordinates (location and viewing direction) along camera rays; (b) generation of densities and colors at the sampling points using NeRF MLP(s); (c) the generation of individual pixel color(s) using in-scene colors and densities along the associated camera rays via volume rendering; (d) the comparison to ground truth pixel colors.

PDF along the ray to guide the location distribution of the sampling points in the fine stage. During the fine stage, sampling is concentrated in areas of higher density identified in the coarse stage, ensuring that the sampling points used for synthesizing the final color are distributed across the object's surface. Subsequently, the sampling points from the fine stage are used to compute the predicted color of pixels, according to the volume rendering integral in Figure 6(c). The training process involves optimizing the parameters of the neural network, minimizing the difference between predicted images and ground truth images for each pixel using a mean squared error (MSE) loss (Figure 6(d)), as shown in (1).

$$L = \sum_{\mathbf{r} \in R} \left\| \hat{C}(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2, \tag{1}$$

where $\hat{C}(\mathbf{r})$ represents the predicted color of the pixel corresponding to the sampled camera ray $\mathbf{r}$, which is integrated by the sampled c along r. R is the batch of sampled rays, and $C(\mathbf{r})$ is the ground truth.

4.1.2 *Construction acceleration*

Traditional implementation of NeRF is plagued by excessively long training durations. Current studies on accelerating NeRF training primarily focus on identifying a more efficient scene representation to replace the parameter-intensive MLP. Point-NeRF [31] employs feature point clouds as an intermediary step in volume rendering. A pre-trained 3D CNN is utilized to generate depth and surface probability γ from cost volumes created from training views, resulting in the generation of dense point clouds. Point-NeRF represents a volumetric radiance field using a neural point cloud, facilitating highly efficient scene reconstruction through optimizations that take only 20–40 min per scene, in contrast to the original NeRF's requirement of over 20 h. Fridovich-Keil et al. [32] introduced the Plenoxels, which represent scenes as sparse 3D voxel grids, with each voxel storing density and spherical harmonic coefficients. This representation can be optimized directly on the voxel grid from calibrated images through gradient and regularization optimization, entirely bypassing MLP training. The voxel training process begins with a lower-resolution dense grid, optimizes and eliminates unnecessary voxels, refines the remaining voxels by upsampling in each dimension, and continues optimization. This method has reduced the training time of NeRF to 12 min.

To achieve a method that can be trained quickly and does not require extensive storage space during the training process, Chen et al. [33] proposed the TensoRF model. Addressing the inefficient use of voxel grids in training by previous methods, TensoRF models the scene's radiance field as a 4D tensor, representing the 3D voxel grid with multi-channel features for each voxel. Subsequently, conventional tensor decomposition algorithms are used in the radiance field modeling process, decomposing the full tensor of the radiance field into multiple compact, low-rank tensor components. Tensor decomposition algorithms can reduce dimensionality and compress data, achieving fast reconstruction speeds of less than 10 min while reducing spatial occupancy during modeling. Müller et al. [34] proposed Instant-NGP, a learnable, parametric multi-resolution hash encoding that is trained concurrently with NeRF's MLPs. Through this parametric approach, combined with advanced ray marching techniques, including exponential stepping, skipping empty spaces, and sample compression, Instant-NGP reduces training time to just a few seconds.

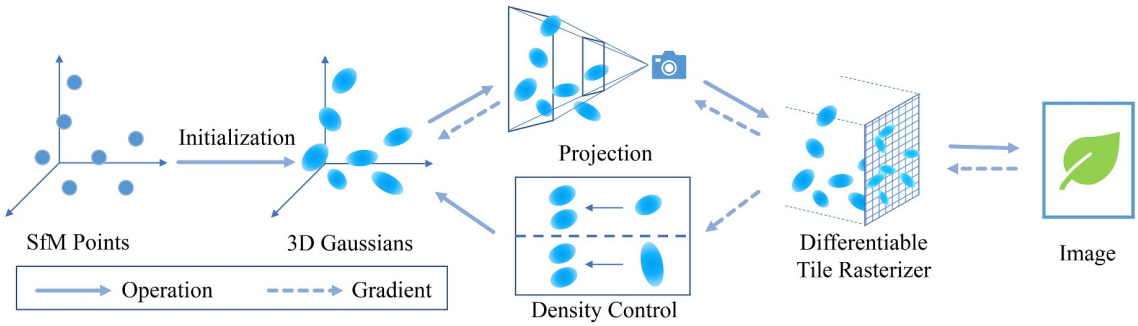

Figure 7 (Color online) 3D Gaussian representation in the work of Kerbl et al. [35]. Optimization starts with the sparse SfM point cloud and creates a set of 3D Gaussians. Then, Gaussians are optimized and adaptively control the density. During optimization, a fast tile-based renderer is first used.

4.2 3D Gaussians

Kerbl et al. [35] introduced an advanced, explicit scene representation with 3D Gaussians to efficiently render complex scenes with a high degree of detail in 2023. The core concept of this 3D Gaussian representation is to depict a scene as a set of learnable three-dimensional Gaussian ellipsoids. Each Gaussian ellipsoid is characterized by its position (center) μ, opacity α, 3D covariance matrix $\sum$, and color c. c is represented by spherical harmonics for view-dependent appearance. The position, size, and orientation properties learn the scene's geometry, while the color and opacity properties learn the photometric properties of the scene.

A Gaussian can be defined by a full 3D covariance matrix $\sum$ defined in world space centered at point μ with

$$G(x) = \mathrm{e}^{-\frac{1}{2}(x)^{\mathrm{T}}\sum^{-1}(x)}. \tag{2}$$

By skipping the third row and column of $\sum$, the 2×2 variance matrix has the same structure and properties. However, covariance matrices have physical meaning only when they are positive semi-definite. The gradient descent cannot be easily constrained to produce such valid matrices, and update steps and gradients can very easily create invalid covariance matrices. To contain the positive semi-definite of the covariance matrices, Kerbl et al. [35] used an ellipsoid to represent a 3D Gaussian, because the covariance matrix $\sum$ of a 3D Gaussian is analogous to describing the configuration of an ellipsoid. Given a scaling matrix S and rotation matrix R, the $\sum$ can be computed with

$$\sum = RSS^{\mathrm{T}}R^{\mathrm{T}}. \tag{3}$$

Since the Gaussian ellipsoid has a density property, 3D Gaussian can represent not only the effect produced by light interacting with the scene's surface but also the effect of light passing through an object. Compared with NeRF, 3D Gaussian can efficiently and accurately represent geometric shapes and appearance properties. Figure 7 shows the overview of 3D Gaussian representation. 3D Gaussian overcomes the limitations of volumetric rendering methods and enables a more flexible and adaptive representation of 3D objects. Additionally, 3D Gaussian can be used to render various visual effects such as depth of field and soft shadows, making it an important representation for VR applications.

4.2.1 *3D Gaussian construction*

The process of 3D Gaussian construction is essentially a training process. This process takes a set of images with corresponding camera poses as input and outputs a set of 3D Gaussians. All the properties of 3D Gaussians can be learned and optimized by gradient back-propagation.

3D Gaussian is first initialized with the SfM points, which are a set of sparse points. Then, the Gaussians are rendered using differentiable rasterization to generate the rendered image. Next, the loss is computed based on the rendering and ground-truth images, and the properties of Gaussians are optimized based on the loss. The loss L contains $L1$ loss and D-SSIM loss shown as

$$L = (1-\lambda)L_1 + \lambda L_{\text{D-SSIM}}. \tag{4}$$

Due to the ambiguities of 3D to 2D projection, geometry may be incorrectly placed. Thus, an adaptive density control algorithm is applied to create, destroy, or move geometry if it has been incorrectly positioned. This

algorithm has three components: clone, split, and prune. This algorithm first detects under-reconstruction and over-reconstruction regions based on view-space positional gradients. Then, for Gaussians in the under-reconstruction region, it clones the Gaussians by creating a copy of the same size and moving it in the direction of the positional gradient. For Gaussians in the over-reconstruction region, it replaces them by two new ones and divides their scale. To moderate the increase in the number of Gaussians, this algorithm also removes Gaussians with less density.

4.2.2 *Construction acceleration*

One of the great advantages of 3D Gaussian representation is that the integral of a 3D Gaussian along a certain axis is a 2D Gaussian; thus one sample is needed for one 3D Gaussian, which allows 3D Gaussian Splatting to have higher performance than NeRF. However, in construction, the depth sorting for each pixel during rasterization is time-consuming. To accelerate construction, Kerbl et al. [35] proposed a tile-based parallel computation method. To avoid the computational cost of deriving a Gaussian for each pixel, this method shifts the accuracy from the pixel-level to the patch-level detail. Specifically, it first divides the image into multiple non-overlapping patches called "tiles." Each tile consists of 16×16 pixels. Then, it determines which tiles intersect these projected Gaussians, duplicates the Gaussians that cover more than one tile, and assigns each copy an identifier (i.e., a tile ID) for the tile. Next, it combines the respective tile IDs with the depth values obtained from each Gaussian view transform and gets an unsorted list of bytes, where the high bit represents the tile ID and the low bit represents the depth. Then, it sorts the list, and the sorted list can then be used directly for alpha synthesis. To speed up the construction, each tile and pixel is rendered independently, so the process is accelerated in parallel. In addition, the pixels of each tile have access to the common shared memory and maintain a uniform read order, thus improving the efficiency of parallel execution of alpha synthesis.

In order to represent large and dynamic scenes accurately, millions of Gaussians are needed to represent the scene, which still takes a significant amount of time to build, even with the tile-based parallel computation method described above. Therefore, it is crucial to reduce the modeling time while maintaining the modeling accuracy. There are three main directions to improve the construction efficiency.

The first direction involves reducing the number of 3D Gaussians, i.e., pruning insignificant 3D Gaussians. Lu et al. [36] proposed Scaffold-GS, which constructs efficiently while maintaining comparable modeling quality. Scaffold-GS utilizes the underlying scene structure to guide the pruning of overextended Gaussian spheres. It uses initialization points from the motion structure to build a sparse mesh of anchor points and attaches a set of learnable Gaussian balls to each anchor point. The properties of these Gaussians are predicted on the fly based on specific anchor point features. In addition, a strategy guided by the aggregated gradient of the neural Gaussian is used to add anchor points. An additional volumetric regularization loss term is added to encourage Gaussian minimization and minimize overlap.

The second direction focuses on improving the construction efficiency by compressing 3D Gaussian properties. Katsumata et al. [37] proposed to reduce the time required to construct the 3D Gaussians for dynamic scenes. It categorizes Gaussian parameters into time-invariant and time-varying parameters. The former contains position and rotation parameters, which are estimated by Fourier and linear approximations, respectively. This approach effectively reduces the time during construction compared to constructing each parameter at each time step. In addition, the flow information overcomes the ambiguity between consecutive frames with different time steps by means of a loss term. Fan et al. [38] proposed the LightGaussian to improve construction efficiency. It first evaluates the global importance of each Gaussian based on its contribution to each pixel in all training views. The computed score is then used to remove unimportant Gaussians. In addition, it reduces the degree of the spherical harmonic coefficients by data distillation and quantizes the coefficients of trivial Gaussians. In addition, the positional parameters are compressed by a lossless octet-based algorithm, and the remaining attributes are saved in a half-precision format.

The third direction is to model the scene with different levels of precision and select different levels according to the desired modeling quality during construction. Kerbl et al. [39] proposed the Hier3D-Gaussian, which models the scene with a hierarchy of 3D Gaussians and uses a divide-and-conquer strategy for accelerating construction. Hier3DGaussian first divides the 3DGSs into multiple axis-aligned bounding boxes (AABBs) containing only one Gaussian based on its spatial location by the division method and uses these AABBs containing only one Gaussian as leaf nodes, and takes each intermediate AABB in the division process as an intermediate node. The process of division is top-down, and when the division process is complete, a multi-layered binary tree is constructed. Each node of the tree is an AABB, and the leaf node contains only 1 Gaussian. Then, Hier3DGaussian calculates the Gaussian properties of each intermediate node based on the Gaussian properties of the leaf node from the bottom upward. After that, a hierarchy of 3D Gaussians is constructed, with the initial 3D Gaussian at the

Table 3 Summary of studies related to content rendering.

	Material & lighting			
	Entangled		Disentangled	
	Novel view synthesis (NVS)	NVS with edition	Material model	Light transport
Mesh rendering	N/A	N/A	[112–114, 116, 118–120, 124, 128, 131]	[115, 124, 135–149, 151–154]
NeRF rendering	[30–34, 62–65, 69–72, 74, 75, 79, 84, 88, 91–93, 95, 111, 186, 245, 249, 377, 388–391]	[83, 86, 87, 280, 312]	[89, 90, 205]	[89, 90]
Gaussian Splatting	[35, 36, 38, 39, 97–100, 102, 104–108, 210, 225, 306]	[101, 103, 314, 379]	[109, 110]	[109, 110]

bottom layer and the intermediate 3D Gaussian obtained by aggregation in the middle layers. For large scenes, Hier3DGaussian first divides the scenes into chunks and then constructs 3D Gaussians for each chunk in parallel.

5 Content rendering

3D content rendering is the key to achieving realistic visual effects in VR systems, which can significantly enhance users' immersion and enable them to experience near real-world sensations in the virtual world. Deep learning methods improve visual quality by optimizing the rendering pipeline, achieving significant breakthroughs in detail and realism. At the same time, these methods can also improve rendering efficiency, which enables the virtual reality system to respond to user interactions at a higher speed, ensuring the smoothness and interactivity of the virtual reality experience and enabling users to freely explore in a more realistic and dynamic virtual environment, thus obtaining a deeper level of immersive experiences. We present a taxonomy of content rendering methods in Table 3, organized by two dimensions: rendering method (vertical axis) and appearance models (horizontal axis). In terms of rendering methods, we consider three types: mesh rendering, NeRF rendering, and Gaussian Splatting. Regarding appearance models, there are mainly two schemes: the material and lighting are decoupled or entangled, where the former line mainly focuses on the novel view synthesis (NVS) task or with further editing, and the latter group aims at the reconstruction and relighting task, requiring to model both the materials and the light transport. Note that we did not include mesh-based methods that entangle material and lighting since they are less relevant to the topic of this paper. In the following subsections, we first review these three rendering methods and then discuss material representation, operations, and light transport computation. Finally, we discuss rendering post-processing.

5.1 Differentiable rendering

Traditional rendering approaches generate images by simulating light transport in 3D scenes. In contrast, differentiable rendering not only produces rendered images but also computes their derivatives with respect to (w.r.t.) various properties. Here, the properties include geometries, materials, camera pose, and image space properties. When derivatives of different properties are available, they can be used for gradient-based optimization or backpropagation in neural networks, enabling various applications, such as content generation, 3D reconstruction, appearance capture modeling, and inverse optical design in virtual reality. While some previous studies focus on designing a particular differentiable pipeline for the downstream task, others target general-purpose differentiable rendering systems or methods. In this section, we focus on the latter category.

Differentiable rendering can be applied for different geometry representations (e.g., mesh, volume, or implicit representations) or different rendering methods (e.g., rasterization or ray tracing). At the core of differentiable rendering on different representations or rendering methods is how to handle discontinuous properties. In this paper, we categorize these methods into two groups, depending on their primary purpose: image quality or performance.

5.1.1 *Physically based differentiable rendering*

Previous studies introduce differentiation into the Monte Carlo raytracing, path tracing, or more advanced volumetric light transport algorithms. These methods can model complex light transport, like global illumination and occlusion, at the cost of introducing noise and long converging time.

Li et al. [40] presented differentiable Monte Carlo ray tracing by edge sampling that directly samples the Dirac delta functions introduced by the derivatives of discontinuous integrals. Later, Loubet et al. [41] proposed reparameterization to handle the non-continuous visibility computation, which is also used in Mitsuba 2 [42]. While its solution significantly improves the performance, it introduces bias and is limited to unidirectional path tracing.

Xu et al. [43] extended the hemispherical-integral reparameterization [41] into the path space, allowing advanced Monte Carlo rendering methods. Zhang et al. [44] established the differential path integral formulation, including interior and boundary components, which can be estimated by both differentiable unidirectional path tracing and bidirectional path tracing, achieving unbiased and efficiency on complex light transport. The differentiable formulation is later generalized to handle participating medium [45] and implicit surfaces [46] by introducing new Carlo estimators for sampling implicitly specified discontinuity boundaries.

Physically based differentiable approaches rely on Monte Carlo based estimators, leading to the typical noise issue. To reduce noise, advanced sampling techniques or other strategies (e.g., temporal reusing or blurring) have been introduced. Different from forward Monte Carlo rendering, values of the derivative function might be negative, causing difficulties in sampling. For that, Zhang et al. [47] introduced an antithetic sampling of BSDFs and light-transport paths, allowing significantly faster convergence. Recently, reservoir-based spatiotemporal resampled importance resampling (ReSTIR) has also been introduced into gradient computation by Wang et al. [48] and Chang et al. [49] to reduce the noise of gradient images. Besides sampling, Fischer et al. [50] proposed to convolve the high-dimensional rendering function with an additional kernel that blurs the parameter space, together with two Monte Carlo estimators for efficiently computing plateau-reduced gradients, showing benefits on complex light effects, such as caustic or global illumination. To improve convergence in inverse rendering optimizations, Xing et al. [51] proposed to compute derivatives on visible 3D geometric points rather than on pixels and compute the 5D RGBXY derivatives (3D for RGB color and 2D for projected screen-space position) w.r.t. scene parameters, leading to superior convergence. To handle complex light transport with specular effects in inverse optimization, Xing et al. [52] introduced extended path space manifolds for path derivative computation.

Besides reducing noise, memory cost is another issue for differentiable rendering. For that, Vicini et al. [53] proposed path replay backpropagation, which recovers quantities needed for reverse-mode differentiation using the invertibility of the local Jacobian at scattering event, leading to constant memory and linear (in terms of the scattering event) computation time is linear in the number of scattering events (i.e., just like path tracing).

5.1.2 *High performance differentiable rendering*

While the physically based differentiable rendering approaches aim at high-quality derivative computation, the other line of studies' purpose is high performance by applying differentiation into the rasterization pipeline (e.g., vertex shading, primitive assembly, geometry shading, etc.) at the cost of missing some light effects (e.g., occlusion, or global illumination).

Early research in differentiable rendering primarily relied on methods such as local derivative approximations, smooth rasterization, and probabilistic rasterization. Among this group, OpenDR [54] is the first general-purpose differentiable renderer, despite its simplified shading model. Kato et al. [55] introduced a smooth rasterization by smoothing the influence of vertex movements on pixels and deriving an approximate method for calculating the pixel-to-vertex coordinate derivatives while maintaining the forward rendering unchanged, leading to the risk of inconsistency between the rendered image and the gradient. Liu et al. [56] proposed a probabilistic rasterization method that modifies the rasterization and z-buffering steps using probability distributions. Their approach ensures that the forward rendering process is fully differentiable. However, the blur operation might lead to unexpected transparent surfaces for opaque surfaces. Genova et al. [57] utilized barycentric coordinate interpolation of vertex attributes to compute pixel colors, performing well in scenarios with smooth and simple occlusion boundaries but struggling with complex, overlapping occlusion boundaries. More efforts [58–60] have been made by exploring efficient and approximate methods for rendering and gradient computation.

While it is challenging to keep scalability, flexibility, and other features (e.g., antialiasing) for this group of methods, Nvdiffrast [61] is an exception, which supports a full package of operations, including rasterizing large numbers of triangles, attribute interpolation, filtered texture lookups, user-programmable shading, and geometry processing, although global illumination is not supported. Therefore, it is still being used in many practical applications.

5.2 NeRF rendering

The basic rendering pipeline of the trained NeRF representation is as follows. (1) For a given 6-DoF camera pose, generate camera rays corresponding to each pixel in the two-dimensional screen space. (2) For each camera ray, generate a set of sampling points along the ray. (3) For each sampling point, input the corresponding ray direction (θ, ϕ) and the 3D position (x, y, z) into the trained NeRF MLP to calculate the color c and density σ. (4) Using the volume rendering formula as illustrated in Eq. (5), accumulate the color and density of all sampling points along the ray to compute the color of the pixel corresponding to the ray. (5) Repeat steps (2)−(4) for all camera rays to generate the final rendered image.

Rendering equation. To compute the color of each pixel corresponding to each ray $\mathbf{r}$ passing through the scene, classic ray marching and volume rendering formulas are used to describe how light traverses and interacts with the scene's surface shown as

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i \delta_i))\mathbf{c}_i, \tag{5}$$

$$T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right), \tag{6}$$

where $\hat{C}(\mathbf{r})$ represents the predicted final color of the pixel corresponding to the camera ray $\mathbf{r}$, σ and $\mathbf{c}$ represent the volume density and the color of the sampling points predicted by the MLP, and δ represents the interval between adjacent sampling points. T_i represents the accumulated transmittance, describing the occlusion conditions when the ray steps to the ith sampling point. Additionally, some methods often refer to the product terms beyond c as the importance weight w_i,

$$w_i = T_i(1 - \exp(-\sigma_i \delta_i)), \tag{7}$$

representing each sampling point's contribution to the final color $C(\mathbf{r})$.

5.2.1 *Acceleration techniques*

The design of efficient computational processes is crucial for the application of NeRF representations in VR. In the process of image generation by traditional NeRF [30], each pixel necessitates approximately 200 forward predictions by an MLP deep learning model. Although the computational scale of a single calculation is modest, the cumulative computational load for rendering an entire image through per-pixel calculations becomes substantial. Currently, mainstream studies to accelerate NeRF inference primarily focus on model baking, which precomputes NeRF and stores the results in efficient explicit data structures, thus avoiding dense MLP inference during real-time rendering. Additionally, techniques that accelerate ray marching, such as early termination and skipping empty spaces, are also utilized to further increase the inference speed of NeRF.

DIVeR [62] performs deterministic ray sampling on a voxel grid, generating an integrated feature for each ray interval (defined by the intersections of the ray with voxels), which is then decoded by an MLP to produce the density and color for the ray interval. This effectively inverts the traditional sequence between volumetric rendering and MLP inference. Experimental results indicate that this method surpasses others, such as PlenOctrees and FastNeRF, in terms of rendering quality while maintaining a comparable rendering speed. SNeRG [63] pre-calculates diffuse color, density, and specular reflection feature vectors and stores them in a sparse voxel grid. At the time of inference, these feature vectors are processed through a lightweight MLP to produce specular reflection colors, which are subsequently alpha-blended with the specular colors along the ray, culminating in the generation of the final pixel color. SNeRG is 3000 times faster than the original NeRF while achieving higher quality results. PlenOctree [64] trains a spherical harmonics NeRF, referred to as NeRF-SH, which predicts the spherical harmonics coefficients of the color function instead of directly predicting the color function itself. Additionally, an octree based on MLP-precomputed spherical harmonics coefficients is constructed. Garbin et al. [65] introduced the FastNeRF technique, which accelerates the rendering time of NeRF to 200 FPS by caching the inferred color and density results of the scene within a dense grid. This method also leverages a hardware-accelerated ray tracing strategy, enabling it to bypass empty spaces and halt prematurely once the transmittance along the ray reaches saturation.

5.2.2 *Foveated NeRF rendering*

Foveated rendering is a pivotal technique for VR and AR applications aimed at reducing computational load and enhancing rendering performance. Potter et al. [66] demonstrated that the human visual system's (HVS) tolerance threshold for visual latency is approximately 13 ms, suggesting that excessive rendering delays in VR can lead to a perceived inconsistency between content and interaction, thereby causing discomfort. The HVS model is closely related to neuroscience and cognitive science. Its core objective is to understand the biological mechanisms of visual information processing based on the characteristics of the human visual system and apply them to technological development. In the human visual system, the distribution of optic nerve projections on the retina is uneven—50% of them are concentrated in the fovea, while the rest are distributed in the peripheral region [67]. This characteristic results in clear perception in the central area and blurred perception in the peripheral area when observing a scene [68], providing insights for accelerating rendering. Foveated rendering achieves acceleration by rendering different image qualities for different regions. The most representative HVS models in foveated rendering

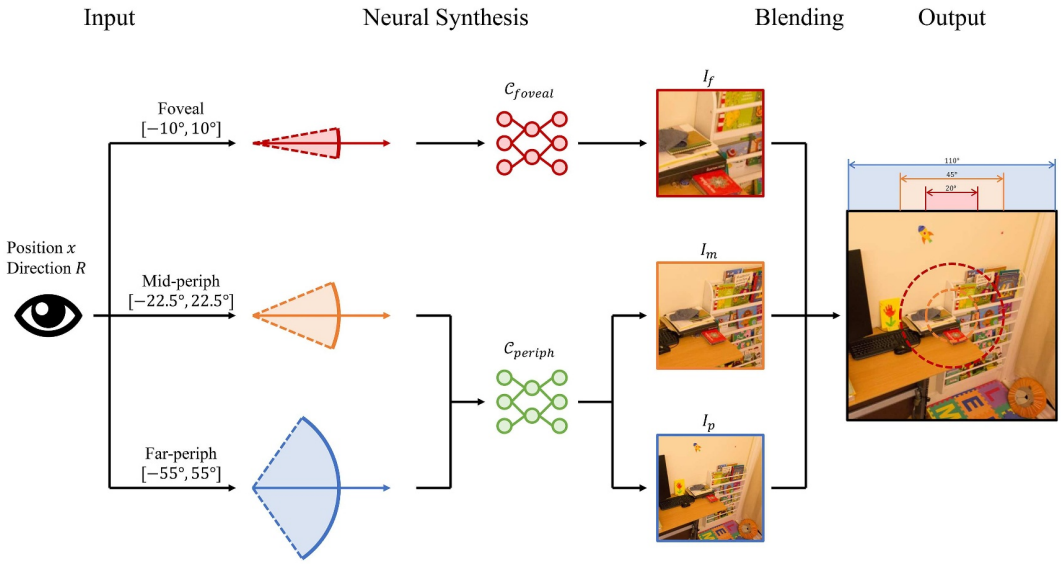

Figure 8 (Color online) Visual acuity adaptive synthesis and rendering mechanism in Fov-NeRF in the work of Deng et al. [69]. Elemental images are synthesized from an egocentric neural representation for the fovea, mid-periphery, and far-periphery. These images are then blended into the final displayed frame.

are the visual acuity model and the contrast sensitivity model. The visual acuity model describes the attenuation of visual acuity as the distance from the central line of sight increases. Deng et al. [69] were the first to combine the NeRF model with the foveated rendering technique, proposing the FoV-NeRF method. Figure 8 visualizes the FoV-NeRF method rendering pipeline. This method employed a concentric sphere coordinate system to represent the 3D radiance field, thereby optimizing the ego-centric view while significantly reducing the inference runtime of the neural network. It derived a spatiotemporal awareness model from the optimized neural scene representation to minimize imperceptible losses in image quality and rendering latency. Shi et al. [70] proposed a scene-aware foveated Nerfs method. It constructed a multi-ellipsoid neural representation to enhance the representation of the neural irradiance field in salient regions of complex VR scenes. It also introduced a uniform sampling-based foveated Nerf framework to improve the foveated image synthesis performance with one-pass color inference. Compared with FoV-NeRF, the synthesis quality and rendering performance are improved. Wang et al. [71] proposed a new NeRF representation based on visual perception, which for the first time, integrates the visual sensitivity and contrast sensitivity models of the human visual system into a NeRF rendering framework. They also adopt a visual-perceptual sampling strategy to allocate computational resources according to the sensitivity of the human eye to the HVS. It achieves better image quality and a higher frame rate than FoV-NeRF.

5.2.3 *Dynamic scenes rendering*

The original NeRF [30] approach was confined to static scenes, rendering it inapplicable in dynamic environments directly. Numerous studies have sought to extend 3D NeRF across the temporal dimension. D-NeRF [72] represents the pioneering work in end-to-end dynamic NeRF, segmenting learning into two modules: the first model discerns spatial mappings between each point in a scene at time t and a canonical scene configuration; the second module regresses the scene radiance emitted by each direction and volume density for a given tuple: at a 3D point (x, y, z) and viewing direction (θ, ϕ), it returns the emitted color c and volume density σ, aligning with traditional NeRF's methodology.

Li et al. [73] proposed a methodology for synthesizing new viewpoints and temporal compositions in dynamic scenes, requiring only a monocular video with a known camera pose as input. This introduced a neural scene flow field that models dynamic scenes as continuous functions of space and time, outputting not only reflectance and density but also the motion of the 3D scene. Park et al. [74] first introduced NeRFies, which similarly utilizes a deformation field to model non-rigid deformations within the scene. Unlike D-NeRF, this method does not utilize temporal sequences as input for predicting deformations. Instead, it employs a latent code to encode the appearance of objects, relying on a latent deformation code to predict object deformations. Building on NeRFies, Park et al. subsequently proposed HyperNeRF [75], which extends canonical space into higher dimensions to address complex topological transformations, adding an additional slicing MLP that describes how to return to the 3D representation using the ambient space coordinates. Xian et al. [76] developed a method to learn spatiotemporal neural irradiance

fields from a single video, applying depth supervision to constrain the time-varying geometry of dynamic scenes at any moment, thereby resolving ambiguities associated with appearance changes in motion-filled scenes. Zhang et al. [77] proposed a differentiable point-based rendering algorithm that achieves efficient new viewpoint synthesis through a differentiable splat-based rasterizer. Starting from a uniformly sampled random point cloud, it learns each point's position and view-dependent appearance. For dynamic scenes, it trains a model for each frame image and uses the model learned in a given frame to initialize the next, thereby reducing the training time required to converge.

Recently, more studies on dynamic NeRF have focused on innovative methods for 4D dynamic scene modeling. Wang et al. [78] introduced Fourier PlenOctrees, achieving real-time rendering for general dynamic scenes. This method models time-varying density and light field functions in dynamic scenes using Fourier coefficients, employing an octree structure to accelerate NeRF inference. NeRFPlayer [79] is a feature-streaming scheme based on a hybrid representation that effectively models dynamic scenes. This method decomposes the 4D spacetime according to temporal characteristics, associating points within the 4D space with probabilities belonging to static, deforming, and new regions, each represented and normalized by an independent neural field. Tensor4D [80] presents an efficient and effective method for dynamic scene modeling, relying on an efficacious 4D tensor decomposition approach to represent dynamic scenes directly as 4D spacetime tensors. It projects the 4D tensor onto three time-aware volumes, then onto nine compact feature planes, stratifying the decomposition of the 4D tensor to capture spatial information that changes over time compactly and efficiently. HexPlane [81] uses six learnable feature planes to explicitly represent 3D dynamic scenes, overcoming the memory challenges of modeling all 3D points in space and time. By projecting points onto each feature plane and then aggregating the six resulting feature vectors, HexPlane computes the spatiotemporal feature vector of points, predicting the color of the point through a minimal MLP, thus achieving new viewpoint synthesis results in dynamic scenes. Fridovich-Keil et al. [82] extended HexPlane to arbitrary dimensions, introducing K-Planes, a white-box model capable of modeling human-perceived radiation fields across dimensions using d-2 planes to represent scenes in d dimensions, such as static scenes ($d = 3$) and dynamic scenes ($d = 4$). This method efficiently models static and dynamic scenes and those with variable appearances, with reconstruction quality competitive with or superior to MLP-based black-box models.

5.2.4 *NeRF controllable editing*

Although neural radiance fields provide a plausible representation of scenes, they do not inherently support editable transformations of shape and appearance. This limitation arises because NeRF models a scene as a radiance-dense field, meaning any alterations to the scene necessitate extensive computational resources and time to recalculate the new radiance field. Enabling controlled modifications to scene representations represents a critical development vector for NeRF, with primary emphases on shape, appearance, and scene composition.

Liu et al. [83] proposed EditNeRF, which allows users to input image conditions, enabling localized edits to an object's appearance, including color and shape. The model comprises a category-specific shared shape network and an instance-specific shape network. The editing process in NeRF is achieved through joint optimization that balances the accuracy and the losses of latent encodings. These encodings permit the NeRF model to control each image's lighting and shading variations, as well as minor content changes in the scene. CodeNeRF [84] learns to decouple the connections between shape and texture by studying separate embeddings. It segregates geometry, appearance, and viewpoint through a fully connected network that maps 3D locations and ray directions to density and RGB values. Objects novel viewpoint images can be reconstructed from a single image, with subsequent edits to shape and texture possible by rendering new perspectives or altering latent codes. However, these shape and appearance latent encodings are typically 2D. GIRAFFE [85] also employs generated latent encodings, predicated on the critical assumption that integrating synthesized 3D scene representations into generative models can render image synthesis more controllable. GIRAFFE separates the scene into background and foreground via MLPs, enabling the isolation of one or more objects from the background, and their individual shapes and appearances, allowing translational and rotational movements within the scene, as well as alterations to camera pose.

Another approach to achieving controllable editing of NeRF is to introduce other models and multi-level feature learning strategies, thereby enabling precise control over individual objects within the scene. Yang et al. [86] introduced a composite model that enables editing multiple objects within a scene. This method adopts a voxel-based approach, learning separate latent representations for each object in the scene. Specifically, the model includes two branches: an object branch that encodes each individual object and a scene branch that encodes the scene's geometry and appearance. The object branch conditions on a learnable object activation code, thus facilitating object-level editing capabilities. DFF [87] leverages existing supervised and self-supervised 2D image feature extractors (such as CLIP-LSeg or DINO) to transfer knowledge into a 3D feature field optimized concurrently

with the radiance field, allowing NeRF to be decomposed into any semantic units. This enables multifunctional scene editing through text and image without the need to retrain the radiance field. CLIP-NeRF [88] integrates the training model with NeRF image synthesis, supporting text and image-driven NeRF editing. This method infers shape and appearance codes from real images, introducing shape encoding and appearance encoding to modify the 3D model's volume and color. Thus, NeRF can extract latent space shifts induced by shape and appearance mapping networks from user-inputted text or images.

Some approaches also decompose the scene into more explicit appearance and light models to achieve more detailed and precise control of lights. Chen et al. [89] extracted shading parameters to reconstruct and relight humans from videos. The space-time varying geometry and reflectance are decomposed from the human body as a set of neural fields. Rudnev et al. [90] first introduced NeRF-OSR to relight outdoor scenes based on NeRFs, allowing for the simultaneous editing of illumination and camera.

5.2.5 *Variants of NeRF*

The original NeRF renders each pixel using only a single ray with a relatively low sampling frequency. This insufficient sampling frequency can lead to the aliasing of high-frequency information in the scene, resulting in blurred and jagged renderings. Additionally, the original NeRF requires dense multi-view images and corresponding camera poses for retraining each new scene, which limits its direct applicability to unseen scenes. NeRF has significant room for improvement in modeling accuracy and in reconstructing from sparse views; therefore, numerous variants of NeRF have been developed.

To enhance modeling accuracy, some researchers have modified NeRF's sampling methods. Barron et al. [91] proposed Mip-NeRF, which transforms the original sampling rays into conical frustums and converts the point samples along the rays into samples within these frustums. This approach effectively performs a filtering operation on the feature values of the sampled points within the frustum before rendering. For computational efficiency, Mip-NeRF approximates the conical frustum using a 3D Gaussian distribution to represent the integral region of the radiance field. Additionally, Mip-NeRF introduces a weighted averaging of positional encodings to obtain integrated positional encoding, allowing the neural network to directly infer the average volumetric density and color of the sampled points within the frustum. By effectively rendering conical frustums instead of rays, Mip-NeRF reduces aliasing artifacts and significantly enhances NeRF's capability to capture fine details. They further proposed Mip-NeRF 360 [92], which reduces mean-squared error by 57% compared to Mip-NeRF and is able to produce realistic synthesized views and detailed depth maps for highly intricate, unbounded real-world scenes.

Regarding sparse-view reconstruction, some studies have introduced a convolutional neural network (CNN) for feature extraction and fusion, significantly reducing the number of training samples required and improving NeRFs generalization ability. PixelNeRF [93] employs a CNN encoder to extract image features, endowing 3D points with generalization capabilities and supporting minimal input. After learning the scene prior, NeRF inference under the condition of one or a few input images proceeds by projecting the generated camera rays onto the image plane, extracting image features for each query point. These features, along with the observation direction and query points, are then fed into the NeRF network to generate density and color. PixelNeRF learns a scene prior from multiple images, enabling new view synthesis from a sparse set of views, potentially as few as a single image. Wang et al. [94] proposed an image-based rendering method called IBRNet, which learns a general view interpolation function that can generalize to new scenes. It relies on selecting multiple views from the training set that are most similar to the target viewpoint direction, using CNNs to extract features from these images. This algorithm achieves comparable reconstruction quality for new scenes with results obtained from short-term fine-tuning, similar to NeRF trained for long durations. MVSNeRF [95] also utilizes pre-trained CNNs to extract 2D image features, which are used to construct a 3D cost volume with geometric perception via the plane sweep method. Subsequently, a 3D CNN is used to extract a 3D neural encoding volume. Finally, the MLP decodes volumetric density and radiance for any continuous position within the encoding volume using trilinear interpolation of neural features combined with physics-based volumetric rendering to construct the NeRF. This method enables high-quality radiance field reconstruction from only three sparse input views and achieves realistic view synthesis from the reconstruction results. For new scenes, MVSNeRF can achieve reconstruction quality similar to NeRF trained for 10 h with just 15 min of fine-tuning.

5.3 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) has 3 main steps. Similar to rasterization, given a specified camera pose, frustum culling determines which 3D Gaussians are outside the camera's frustum. 3D Gaussians outside the given view are not involved in subsequent calculations, thus saving computational resources. Then, similar to the projection of

the mesh in the rasterization process, 3D Gaussians (ellipsoids) are projected into the 2D image space (ellipses) for rendering. At last, for each pixel, all overlapping Gaussians can be obtained based on the distance, and alpha blending is adopted to compute the final color of this pixel.

Rendering equations. Given the viewing transformation W, the 3D Gaussians are first projected to 2D, the projected 2D covariance matrix $\sum^{'}$ is computed with

$$\sum^{'} = JW\sum W^{\mathrm{T}}J^{\mathrm{T}}, \tag{8}$$

where J is the Jacobian of the affine approximation of the projective transformation. This projection process is proven to be accurate [96]. Then, the overlap between projected 2D Gaussians and the pixels is computed based on the projected 2D center and $\sum^{'}$. For the color c of each pixel, the overlapped 2D Gaussians are sorted based on view space depth with a single fast Radix sort, and alpha blending is adopted using

$$c = \sum_{i\in N} c_i\alpha_i \prod_{j=1}^{i-1}(1-\alpha_j^{'}), \tag{9}$$

where c_i is the learned color, N is the sorted list of overlapping Gaussians, and the final opacity $\alpha_i^{'}$ is the multiplication result of the learned opacity α_i based on

$$\alpha_i^{'} = \alpha_i \cdot \exp\left(-\frac{1}{2}(x^{'}-\mu_i^{'})^{\mathrm{T}}\sum\nolimits_i^{'-1}(x^{'}-\mu_i^{'})\right), \tag{10}$$

where $x^{'}$ and $\mu_i^{'}$ are pixel coordinate and center of the projected 2D Gaussian in the projected space.

5.3.1 *Dynamic scene rendering*

3DGS cannot be used to render dynamic scenes directly. It would be beneficial to enhance the capabilities of 3DGS by extending it from representing static scenes to dynamic scenes. One way to extend 3DGS to dynamic scenes is to learn a set of Gaussian representations for each image frame, but this is obviously costly. Learning deformations is more convenient than modeling the scene at each time step.

Some research focuses on constructing the deformation field for all Gaussians. Wu et al. [97] proposed a novel framework for real-time 4D dynamic scene rendering. Their framework first employs a spatial-temporal encoder that utilizes multiresolution K-Planes and MLPs for efficient feature extraction. Then, a compact multi-head MLP is used as a decoder to predict positional deformation, rotation, and scaling, respectively. This method of learning the Gaussian deformation field results in efficient memory usage and fast convergence. Duisterhof et al. [98] introduced MD-Splatting to perform 3D point tracking while synthesizing dynamic new views. MD-Splatting employs a feature encoding technique and learns Gaussian deformations in metric space rather than non-metric normed space. In addition, opacity and scale parameters are not inferred to prevent learning opacity, and scale parameters over time render the Gaussian insubordinate to the exact motion of the point. Rigidity and isometric losses, as well as momentum conservation losses, are incorporated into the trajectory regularization. Yang et al. [99] proposed 4DGS to model space and time as a whole to address the problem of representing and rendering dynamic scenes in general. 4DGS extends the scaling and rotation matrices derived from the decomposition of the original covariance matrices to 4D Euclidean space. A generalized 4D Gaussian representation is derived to achieve a reasonable fit to 4D manifolds while capturing the underlying dynamics of the scene. In addition, 4DGS is able to exploit spherical nonlinear harmonics so that the appearance changes with the viewing angle and the color evolves over time.

Another work is to construct the deformation field by pairing a small number of Gaussians and controlling other Gaussians through these Gaussians. Kratimenos et al. [100] presented DynMF for real-time synthesis of dynamic views. DynMF decomposes complex motions in a given scene into several base trajectories, from which motions are derived for each point, and the trajectories are then predicted by MLP. A shared neural basis for all points produces physically plausible and frame-consistent sequences. To prevent the selection of unnecessary bases, DynMF employs a stronger loss term, forcing each Gaussian to select only a few trajectories.

Physical simulation is also used to control 3D Gaussian deformation. PhysGaussian was introduced by Xie et al. [101] to seamlessly integrate physics simulation into 3DGS to generate novel dynamics and views. PhysGaussian first reconstructs the static scene through 3DGS and regularizes the overly lean Gaussian with optional anisotropy loss. Continuum medium mechanics is modeled through a continuous deformation map over time. The Gaussian

kernel is treated as a discrete particle cloud and deforms simultaneously with the continuum. In order to force the deformed kernels under the deformation map to be Gaussian, PhysGaussian utilizes a first-order approximation to describe the particles undergoing local affine transformations. Additionally, there is an option to fill the internal regions of the object via a 3D opacity field to help render exposed internal particles.

5.3.2 *3D Gaussian editing*

Scene editing is sometimes required in VR applications, and it is not easy to edit the scene directly with 3D Gaussian. Some methods enable editing by segmenting 3D Gaussians. Zhou et al. [102] proposed Feature 3DGS, which integrated 3DGS with feature field distillation from 2D foundation models. Unlike traditional 3DGS, feature 3DGS expands Gaussians with semantic features and constructs a 3D feature field. It first trained the Gaussians with semantic features under the supervision of 2D segmentation models. Then, a lightweight convolutional decoder will be used for upsampling to get high-dimensional features. Feature 3DGS achieved faster training and rendering speeds while enabling high-quality feature field distillation, supporting Downstream tasks like semantic segmentation and language-guided editing. Ye et al. [103] proposed Gaussian Grouping, which extended 3DGS to jointly reconstruct and segment in 3D scenes. Gaussian Grouping augments each Gaussian with a compact identity encoding, allowing the Gaussians to be grouped according to their object instance or stuff membership in the 3D scene. Instead of resorting to 3D labels, Gaussian Grouping supervises the identity encodings during the differentiable rendering by leveraging the 2D mask predictions by the segment anything model, along with the introduced 3D spatial consistency regularization. Besides, a local Gaussian editing scheme was proposed to achieve object removal, inpainting, colorization, style transfer, and scene recomposition. Cen et al. [104] proposed SAGA, which can efficiently segment the corresponding 3D target represented by 3D Gaussians with a 2D visual prompt. SAGA attached a scalegated affinity feature to each 3D Gaussian to endow it with a new property towards multigranularity segmentation. A scale-aware contrastive training strategy is applied for the scale-gated affinity feature learning. It first distills the segmentation capability of the segment anything model (SAM) from 2D masks into the affinity features. Then, it employs a soft scale gate mechanism to deal with multi-granularity ambiguity in 3D segmentation by adjusting the magnitude of each feature channel according to a specified 3D physical scale. SAGA can achieve real-time multi-granularity promptable segmentation and scene editing.

Another approach is to directly transform 3D Gaussian to achieve editing. Liu et al. [105] proposed StyleGaussian, which can instantly transfer any image's style to a 3D scene represented with 3D Gaussian. StyleGaussian has three steps: embedding, transfer, and decoding. Initially, 2D VGG scene features are embedded into reconstructed 3D Gaussians. Next, the embedded features are transformed according to a reference style image. Finally, the transformed features are decoded into the stylized RGB. A feature rendering strategy is first applied to cut the memory consumption significantly and enables 3DGS to render the high-dimensional memory-intensive features. It renders low-dimensional features and maps them into high-dimensional features while embedding VGG features. Then, a K-nearest-neighbor-based 3D CNN is applied to eliminate the 2D CNN operations that compromise strict multi-view consistency.

5.3.3 *Variants of 3D Gaussian Splatting*

Although 3DGS has achieved good results in terms of rendering quality, it still has room for improvement in terms of modeling accuracy, anti-aliasing, and specular object modeling. There are various variants of 3DGS. To improve geometric modeling accuracy, some researchers modified the Gaussian representation. Huang et al. [106] proposed 2DGS to model and reconstruct geometrically accurate radiance fields. Different from 3DGS, 2DGS collapses the 3D volume into a set of 2D-oriented planar Gaussian disks, which provides view-consistent geometry while modeling surfaces intrinsically. Besides, 2DGS adapts a perspective-accurate 2D splatting process utilizing ray-splat intersection and rasterization to accurately recover thin surfaces. The significant advantage of 2D Gaussian over its 3D counterpart lies in the accurate geometry representation during rendering.

Regarding anti-aliasing, some research focuses on the rendering quality at different distances, focal, and scales. Influenced by Mip-NeRF [91], Yu et al. [107] introduced 3D smoothing and 2D Mip filters to solve the blurring problem in 3D Gaussian optimization process. The three-dimensional filter, which originates from the Nyquist-Shannon sampling theorem, is a Gaussian low-pass filter that removes high-frequency artifacts by limiting the frequency of the three-dimensional representation to less than half of the maximum sampling rate that comes from the multiview image. On the other hand, the 2D filter is designed to mitigate aliasing problems when rendering a reconstructed scene at a lower sampling rate. It replaces the screen space expansion filter of 3DGS and replicates the behavior of the box filter during physical imaging. This principled approach is better suited for non-distributed scenes with unknown camera poses and zoom coefficients. Yan et al. [108] proposed a multiscale approach to

Table 4 Summary of studies related to NeRF rendering and 3D Gaussian Splatting.

	Non-real-time (FPS<20)	Weakly-real-time (FPS<90)	Real-time (FPS>90)
Basic (PSNR<25)	[74–76, 79, 83, 84, 86, 94]	[69, 70]	–
Detail (PSNR<30)	[30, 72, 73, 85, 87, 88, 92, 93, 95, 105]	[62–65, 97, 101, 107]	[96, 99, 103, 104, 108, 110]
Visually-lossless (PSNR>30)	[80–82, 91, 102, 111]	[71, 77, 98, 109]	[78, 100, 106]

mitigate the aliasing effect in 3DGS. They argue that a large number of Gaussians mainly cause the aliasing effect to fill in regions with complex 3D details. Therefore, they represent the scene with different levels of detail. Within each level, fine-grained Gaussians smaller than a certain size threshold in each voxel are aggregated into larger Gaussians and then inserted into subsequent coarser levels. These multi-scale Gaussians effectively encode high and low-frequency signals and are trained with the original image and its downsampled counterpart. During the rendering process, the appropriate scaled Gaussians are selected accordingly, resulting in improved quality and rendering speed.

Some researchers worked on improving the rendering quality for the scene with specular objects. A photorealistic rendering framework was proposed by Gao et al. [109]. It utilizes a set of relightable 3D Gaussian points to represent the scene. The surface normals are regularized by the consistency between the rendered normals and the pseudo-normals, where the pseudo-normals are computed from the rendered depth map. Geometric cues are introduced by integrating multi-view stereo cues. This approach uses a simplified BRDF model with additional rendering attributes assigned to each Gaussian. The incident light is divided into local and global components, which are represented by the spherical harmonics of each Gaussian and the shared global spherical harmonics multiplied by a visibility term, respectively. To improve the rendering efficiency and quality, the physically based rendering colors are computed at the Gaussian level with additional regularization terms attached during the optimization process. Jiang et al. [110] proposed GaussianShader to further enhance the realism of scenes with specular features and reflective surfaces. GaussianShader explicitly takes into account light-surface interactions and employs simplified approximate rendering equations for high-quality rendering at a much lower time cost. To accurately predict the normals of a discrete 3D Gaussian, the method uses the shortest axis of the Gaussian ellipsoid as the approximate normal and introduces two additional trainable normal residuals for regularization, one for the outward axis and the other for the inward axis. In addition, the consistency of the normal geometry is achieved by minimizing the difference between the gradient normals derived from the rendered depth map and the normal maps rendered using the previously predicted normals. Ma et al. [111] proposed SpecNeRF to improve 3DGS modeling and rendering results for specular. SpecNeRF aims to enhance the capabilities of NeRF by using 3D Gaussian as a novel orientation encoding. It utilizes a set of learnable Gaussians as the basis for embedding a 5D ray space containing the ray origin and ray direction. As a result, the encoding function can be varied spatially, with the spatial features varying in a manner consistent with the behavior of the specular reflection component. This results in better simulation of reflections and improved realism. SpecNeRF also introduces an initialization phase that involves the refinement of Gaussian parameters to facilitate the joint optimization of Gaussian and NeRF. In addition, monocular normals are utilized in the early training phase to provide a supervised signal for the predicted normals and to mitigate shape-radiance ambiguities.

Table 4 compares NeRF and 3D Gaussian Splatting methods in scene reconstruction quality (categorized as basic/detailed/visually-lossless) and rendering speed (non-real-time/weakly-real-time/real-time). Color-coded results show NeRF methods [30, 62–95] predominantly achieve basic reconstruction with limited real-time capability, whereas Gaussian Splatting [96–111] demonstrates superior performance: 100% achieve detailed or visually-lossless quality (vs. NeRF's 66.67%) and 81.25% attain real-time rendering (vs. NeRF's 23.33%). This efficiency gap is especially noticeable in scenarios with limited computational resources, where Gaussian methods have higher frame rates.

5.4 Material representation and operations

High-fidelity materials are essential for rendering realistic virtual scenes, and bidirectional reflectance distribution functions (BRDFs) are the most commonly used formulation for materials in realistic rendering. A BRDF is defined as a reflectance function of illumination and viewing directions, which is naturally 4-dimension. Sometimes artists create textures by providing various BRDF parameters at different texture coordinates, and those cooperating texture maps make up a spatially-varying BRDF (SVBRDF), which is consequently 6-dimensional. For some real-world texture measurements, there may be no explicit texture parameter maps to define such an SVBRDF. Therefore, a more generalized representation called bidirectional texture function (BTF) is also widely used to define optical reflectance values as a 6D function of spatial and angular coordinates.

However, the high dimensionality of these material spaces makes it challenging to represent and manipulate materials efficiently. Approximate analytical models like SVBRDFs can be fast and user-friendly, but they often suffer from lower accuracy and compatibility. Once the analytical model is chosen, it is difficult to include new complex effects, for example, parallax, subsurface scattering, and anisotropy. In contrast, neural networks can represent complex materials inclusively, while the manipulation and semantic explanation of the neural representations remain challenging.

5.4.1 *Neural BRDF representation*

Real-world material measurements are one of the most important sources for realistic appearances. To efficiently represent and evaluate these materials in both performance and storage, some studies are carried out to use neural networks to compress measured BRDFs. Hu et al. [112] used a convolutional autoencoder to compress BRDF slices into latent vectors and reconstruct them. Zheng et al. [113] used a neural process instead, which is more compact and efficient. Sztrajman et al. [114] used small MLPs to represent each BRDF individually and further compressed these network weights using an autoencoder. Some other work tries to introduce advanced machine learning technologies to achieve a better sampling space, where one can utilize sparser samples or efficiently handle new materials beyond the training group. Recently, Fischer et al. [115] used meta-learning to generalize over different kinds of appearances, Gokbudak et al. [116] also generalized the BRDF representations by a hypernetwork, and can estimate the measured BRDFs from a sparse set of input samples.

Complex materials, such as layered materials and micro geometries, are also challenging for neural networks to deal with. On the one hand, such a dataset is scarce and the diversity of material samples is limited. On the other hand, the neural networks also tend to fit smooth signals rather than high-frequency details, which are significant for the realism of such materials. Kuznetsov et al. [117] focused on complex micro geometries, using a generative model to evaluate specular material with high efficiency. Another group of work focuses on layered materials, using neural networks to fit the precomputed reflectance and achieve efficient and noise-free evaluation. Fan et al. [118] trained a large optimization-based universal network to represent BRDFs, and encode high-quality specularities. Guo et al. [119] leveraged meta-learning for modeling and rendering layered materials, using two networks to encode material appearances and map between representation weights and physical parameters. Recently, TG et al. [120] proposed the use of neural networks to compress more complex appearances like bidirectional scattering surface reflectance distribution functions into implicit representations.

5.4.2 *Neural SVBRDF/BTF representation*

Textures are key for rendering realistic virtual environments, and there are also both analytical approximation and generalized appearance representations. Since SVBRDFs are naturally efficient in evaluating and rendering, the neural representation of SVBRDFs is more focused on and used for the recovery and derivation of SVBRDF maps from captures [121–123], and sometimes the shapes of objects are also involved [124–126]. These studies usually used CNNs to encode the spatial correlation of the tables and introduced prior knowledge in image space to help with the high dimensionality of the texture space.

However, for measured BTFs, there are no explicit parameter maps to apply any priors. Therefore, the research on neural BTF compression introduces different ways of modeling and utilizing the spatial relationship among the BTF texels. The fundamental idea to represent a BTF with neural networks is to treat each texel of the texture as a single BRDF, which is also called an apparent BRDF. Rainer et al. [127] trained individual encoders for each BTF and used latent maps to store compressed vectors, and Figure 9 illustrates its network structure. This representation is further improved by Rainer et al. [128] to a unified model with higher quality and more compact compression. Kuznetsov et al. [129] used an overfitted neural network to compress a BTF into pyramid structures and can handle parallax effects. This work is later further extended by Kuznetsov et al. [130] for curved surfaces. With a spatial-angular decomposition, Fan et al. [131] proposed a neural biplane model to represent BTFs with a universal network, encoding both the spatial and directional information from the texture. Recently, Zeltner et al. [132] presented a complete system to embed SVBRDF/BTF textures and rendered them with real-time performance, thanks to the prior transformations and sampling algorithm.

5.4.3 *Neural material manipulation and operation*

The manipulation and operation of the neural representations in the latent space are widely studied in previous work, and by different means, some researchers managed to achieve semantic editing and manifold interpolation of these materials. One classic approach is to find the mapping between physically based parameters and the latent

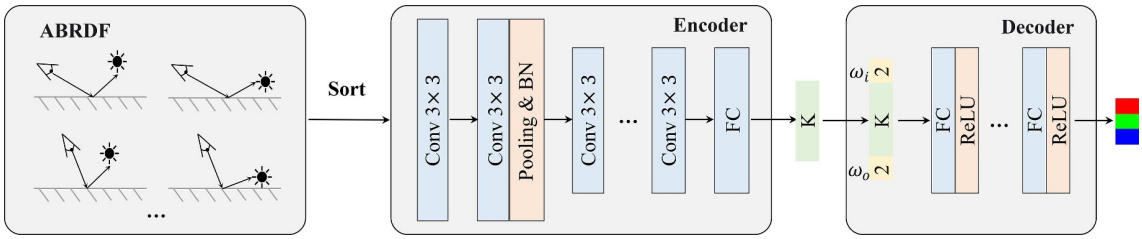

Figure 9 (Color online) The structure of the material representation network in the work of Rainer et al. [127]. Specifically, these studies sample per-pixel apparent BRDFs from the spatially-varying material, encoding them into a latent space, and will be further decoded by a decoder with query directions as inputs. Based on this kind of fundamental pipeline, a series of designs for the encoders and decoders is proposed to achieve different goals and reflect different insights.

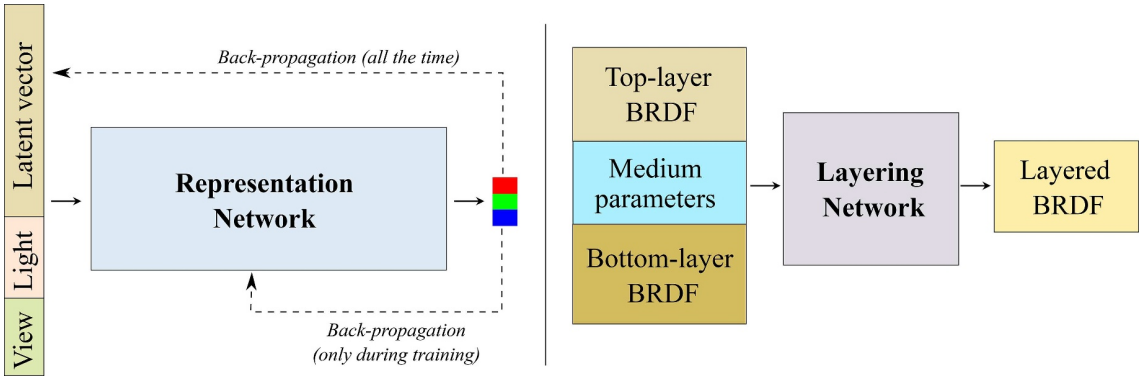

Figure 10 (Color online) Fan et al. [118] introduced a generalized network to achieve high-efficiency layering operation of materials by mapping the latent vectors of layered material to the encodings of their single-layer components and physical parameters. The comparison between the predicted resulting layered BRDFs and the ground truth is shown, indicating the network can also deal with multiple layering operations.

vectors. For example, Hu et al. [112] used an MLP to map physical parameters into the latent space and, therefore, can perturb the material by feeding different parameters. Another way to achieve latent space manipulation is to find mappings within the latent space. For example, Zheng et al. [113] trained several individual classifiers to find some semantically meaningful directions in the latent space and then shifted the latent vectors along these directions. The interpolation of the weights of neural networks for materials is also studied by Sztrajman et al. [114].

Apart from semantic editing and interpolation, some researchers also focus on the advanced operations of materials in the latent space. Fan et al. [118] achieved the layering operation in the latent space using a pretrained layering network by feeding the network with the latent vectors of component materials and some physical parameters as presented in Figure 10. The layering operation is also studied by Guo et al. [119], who use a meta-learning framework to directly predict the resulting materials. Sztrajman et al. [114] computed the CDF for a proxy BRDF instead and trained a network to predict the proxy BRDF parameters. Fan et al. [118] also supported importance sampling for neural materials but directly computed the sampling results from the input latent vector by a sampling network.

5.5 Light transport computation

While the materials in the previous section show objects' 'local' appearance, the light transport across the entire scene is in charge of the 'global' appearance, or so-called global illumination (GI). GI is formulated by the rendering equation (RE) [133] for surface objects and by the radiative transfer equation (RTE) [134] for objects made of volume. As both RE and RTE do not have any analytical solutions, existing approaches rely on Monte Carlo-based techniques to solve them. At the core of these approaches is noise reduction by better path sampling, caching, or control variates. Recently, neural networks have been introduced into these techniques. As path sampling is a typical and active research direction, we discuss the studies related to neural path guiding (or sampling) first, and then we show the other surface rendering techniques with neural works. Finally, we review neural-based participating media rendering approaches.

5.5.1 *Neural path guiding*

The key to path tracing is sampling a path to connect the camera and the light sources. The typical solutions for sampling a path are BRDF importance sampling, light importance sampling, or their combination, or multiple

importance sampling (MIS). While MIS improves the robustness of direct lighting in terms of varying frequency materials or light sources, it fails for indirect lighting, particularly for complex effects, including caustics and glossy indirect illumination. At this point, path guiding is introduced into path sampling, which also considers the global lighting distribution in the scene. The basic idea of path guiding is to store the radiance distribution of each point in the scene and use this distribution to guide the path sampling direction. However, the radiance distribution has five dimensions (three in the spatial domain and two in the angular domain), raising difficulties in storing. Neural networks have shown their powerful capability for representing the spatial-varying distribution of scattered radiance. We classify the related approaches into online learning (or scene-dependent) and offline learning (or scene-independent).

Online learning. One group of path-guiding approaches is online learning, where the radiance distribution is learned in a specific scenario. Müller et al. [135] used an online-learned neural network for an important sample. Despite the high accuracy of the learned distribution, expensive infer time is required for sampling. Zheng et al. [136] introduced RealVNP to warp the primary sample space to obtain desired densities and sample these desired densities rather than uniform primary space sampling. Recently, online learning-based path guiding approaches [137, 138] have also been proposed, using a single MLP to learn the continuous distribution of the scattered radiance product, represented as spherical Gaussians or normalized anisotropic spherical Gaussian mixtures, leading to lower sampling variance. Besides the above studies, the neural network has also been introduced for complex luminaries sampling [139] by learning a coarse, low-resolution distribution with limited accuracy. Wang et al. [140] introduced reinforcement learning in many-light sampling to reduce variance.

Offline learning. In contrast to the online learning category, the offline learning-based approaches learn the distributions with large datasets and can be applied to general scenes. Bako et al. [141] proposed to train a generative adversarial network to reconstruct the desired sampling distribution from the local neighborhood of samples. Similarly, Huo et al. [142] proposed reconstructing the first-bounce incident radiance field with a CNN-based network and sample with the deep reinforcement learning-based network. Unfortunately, these methods are designed for first-bounce sampling only, limiting their supporting GI effects. Later, Zhu et al. [143] introduced photons into path guiding to enable arbitrary bounce sampling. After that, Zhu et al. [144] further improved it by combining photon and path for robustness and including a quad-tree representation of incident radiance distribution using nearest photons to reduce memory consumption.

5.5.2 *Other neural surface rendering techniques*

Besides leveraging neural networks for path sampling, it can also benefit other rendering techniques (e.g., radiance caching and control variates), which aim at accelerating rendering or rendering approaches (e.g., photon mapping).

Ren et al. [145] proposed the first neural network for GI by treating the GI as a radiance regression function, which maps the surface attributes to indirect illumination values. Their method is able to achieve real-time rendering at the cost of expensive precomputation and training. Recently, Gao et al. [146] extended the above method to dynamic area light, together with other techniques, like positional encoding for high-frequency effects, leading to a higher quality. In contrast to offline training, Müller et al. [147] proposed online training for radiance caching by learning the radiance distribution with a single small neural network and leveraging it for real-time rendering, achieving obvious variance reduction. Small neural work and high-performance implementation lead to efficient network inference. Similarly, neural networks have also been leveraged to learn the norm of the residual of the rendering equation by Hadadan et al. [148]. As it is similar to radiosity-based rendering methods, it is also called neural radiosity.

Control variates have been used for noise reduction in Monte Carlo integration. Its key idea is that the integral of an original function is expressed as the known integral of a simpler function, together with the Monte Carlo estimate of the difference to the original integrand. When the difference function is closer to a constant, the resulting variance will be low. Müller et al. [149] introduced neural networks into control variates by learning the control variate, the integral of control variates, and the probability density function of the difference function, leading to obvious noise reduction.

Neural networks have also been applied to other rendering methods, like photon mapping. Usually, photon mapping is used for rendering caustic effects, which require a large amount of photons to achieve the sharp features of caustics. Zhu et al. [150] proposed a deep neural network to predict a kernel function to aggregate photon contributions, leading to obvious photon count requirements to achieve a similar quality.

5.5.3 *Neural participating media rendering*

Different from surfaces, rendering participating media becomes even more challenging due to their complex light effects. For example, light might bounce several times before leaving the medium, leading to a long path. The light transport in a participating medium is formulated by the radiative transfer equation. Solving RTE with Monte Carlo sampling leads to noisy renderings due to the high variance of the sampled long paths. To address this issue, neural networks have also been introduced into participating medium rendering to reduce variance.

Starting from Kallweit et al. [151], the neural network is used to predict the radiance of the participating medium called the radiance prediction neural network (RPNN). Although they encode the shape with stencils, their method cannot handle complex shapes. Upon RPNN, Hu et al. [152] further introduced multiple features for RPNN and decoupled the high-frequency and low-frequency effects, leading to a lighter network, achieving a real-time frame rate. To avoid modeling the shapes explicitly, Leonard et al. [153] relied on sphere tracing and exploited a sequence of conditional variational auto-encoders to model the contributions of all possible paths between two points inside a spherical region. Different from previous studies, Ge et al. [154] applied the neural network to represent multiple scattering of arbitrary homogeneous infinite participating medium directly, without considering the shapes and achieving an interactive frame rate. Although it works for thick medium, it exhibits noticeable differences for thin medium. Unlike the models that train neural networks for the prediction of radiance, Vicini et al. [155] proposed a neural network to sample the exit point to fit the path-tracing framework, where the shapes are encoded by the first-order approximation.

Although hair differs from the participating medium, rendering hair also has the same problem: the long path due to multiple scattering among fibers. KT et al. [156] accelerated hair rendering by learning a small MLP to represent the multiple scattering or high-order scattering online, leading to obviously reduced variance.

5.6 Rendering post-process

After obtaining the advanced material modeling and computing the light transport faithfully, the Monte-Carlo ray-tracing rendering engine is used to render photo-realistic images. However, obtaining noise-free high-resolution images via this method is time-consuming, so rendering post-process techniques are necessary to reduce the rendering cost and improve the rendering quality, mainly including denoising, super-resolution, and frame interpolation/extrapolation.

5.6.1 *Denoising for Monte Carlo rendering*

Monte Carlo (MC) ray tracing comes with significant variance at low sample counts, and denoising is needed as a substantial post-process. These methods are divided into offline and real-time algorithms based on execution time.

Offline denoising. The early studies got inspiration from traditional filter-based methods (e.g., cross bilateral and cross non-local means filters), and they focused on predicting an optimal filter kernel for denoising. Kalantari et al. [157] introduced the neural network for MC denoising. Their method predicts a filter via a multilayer perceptron neural network and conducts filtering to produce denoised images. Bako et al. [158] introduced a CNN model to predict local weighting kernels to filter pixels from their neighbors. Vogels et al. [159] further improved denoising by several task-specific modules and proposed an asymmetric loss to preserve details. Some advanced methods utilize the neural network to predict noise-free images directly and focus on task-specific architecture or learning strategy design. Chaitanya et al. [160] proposed a recurrent neural network (RNN) model considering temporal coherence for interactive renders, which almost runs at real-time rates. Xu et al. [161] introduced adversarial learning into MC denoising for the first time and designed a novel conditioned auxiliary feature modulation method that better utilizes feature information at the pixel level. Yu et al. [162] introduced the self-attention mechanism into MC image denoising, which effectively involves the auxiliary features in the denoising process. Back et al. [163] introduced self-supervised learning into MC denoising and proposed a post-processing network that improves the performance of supervised learning denoisers. They designed a self-supervised loss that guides the post-correction network to optimize its parameters without relying on the reference. The above methods take noisy rendered images as the input, and the sample-based methods consider the information from a single sample. Gharbi et al. [164] presented the first CNN model that can directly learn to denoise from samples. Since samples include more information, the method produces higher quality even with only a few samples. Offline denoising techniques can present exquisite final images. Although it is still difficult to run in real-time and cannot be directly applied in VR applications, the denoising ideas of some algorithms are still very worthwhile. Subsequent research can continue to optimize computational efficiency by considering dynamic and temporal information and designing intelligent filters.

Real-time denoising. Another group of methods can run at real-time rates and thus support online applications. Fu et al. [165] facilitated the U-shape kernel-prediction network with a sparse auxiliary feature encoder, which focuses solely on changed regions and reuses the history features in other regions, reducing 50%–70% consumption without apparent performance drops. Işık et al. [166] designed an MC denoiser that runs at interactive rates, consisting of a filtering algorithm that uses pairwise affinity to learn iteratively-applied 2D dilated kernels and a temporal aggregation mechanism that uses the same pairwise affinity to improve the temporal stability of MC denoising significantly. Balint et al. [167] proposed a pyramidal filter with learnable partitioning and upsampling stages, leading to considerable improvements. Hofmann et al. [168] designed the open image denoiser, enabling combined volume and surface denoising in real time and outperforming current denoisers in scenes containing both surfaces and volumes.

5.6.2 *Super resolution*

Super resolution (SR) aims to increase the spatial resolution of images or video frames. In real-time rendering, these methods are mainly for reducing the computational cost by rendering at a lower resolution and then upsampling to the native resolution. Xiao et al. [169] introduced the deep-learning approach for high-quality upsampling of rendered content. Their method utilizes the available information (e.g., depth, motion vector) across multiple frames and contains a reweighting mechanism to filter out invalid pixels. Guo et al. [170] designed a classifier, and the network uses the classification results to blend the current frame with the warped last frame via a learned weight map to get the supersampling results. They also developed dedicated loss functions to mitigate ghost artifacts. Yang et al. [171] considered the real-time applications on compute-limited hardware. They reuse only one previous frame and use a sub-pixel sample pattern to maximize efficiency and preserve details. Besides, they proposed a novel metric, IF-SSIM, to evaluate the temporal stability of a video quantitatively and a public dataset, GameVideo57, which contains 57 rendered videos and auxiliary buffers. Zhong et al. [172] utilized high-resolution auxiliary G-buffers as additional input and introduced an efficient and effective H-Net architecture to align and fuse features at multi-resolution levels. Several commercial software packages are available to support SR, including DLSS1 [173], FSR1/2 [174], and XeSS [175].

5.6.3 *Frame interpolation/extrapolation*

Frame interpolation/extrapolation aims to increase the frame rate by interpolating intermediate or extrapolating subsequent frames between successive input frames. Guo et al. [176] presented a robust hole-marking strategy to classify the region for inpainting and shading prediction. They also utilize lightweight gated convolutions to enable fast inference. Briedis et al. [177] considered the cases when the motion vectors are not valid and utilize a cost volume built from input frames and auxiliary feature buffers (e.g., albedo and depth) to obtain the optical flow as motion representation. Wu et al. [178] designed a learnable motion vector, which offers more robust motion tracking. They developed a feature streaming network, dubbed FSNet, to enable adaptive frame prediction to serve diverse applications on demand. Briedis et al. [179] designed a kernel-based interpolation model consisting of an attention-inspired mechanism to fill holes in warping keyframes and an adaptive interpolation strategy to achieve better results for a given render budget. Wu et al. [180] built a unified pipeline for both SR and extrapolation, which contains a lightweight G-buffer guided warping module to obtain a good initialization and a flow-based Refinement network to generate high-quality results. He et al. [181] adopted a unified context and a shared neural network to achieve high efficiency and designed a reshading random masking and efficient reshading module to improve the performance. Their method also supports joint SR and extrapolation. Some advanced software supports joint SR and frame interpolation/extrapolation, including DLSS2/3 [173] and FSR3 [174].

6 Content generation

3D content is the dominant content form in VR. Most VR application developments begin with constructing 3D objects or 3D scenes as the foundation for the subsequent rendering and simulation processes. The recent progress of deep generative models provides a new way to generate 3D content conditioned on text or 2D images, which significantly reduces the workload of artists or the burden of capturing the process in 3D reconstruction.

6.1 Generative models

In machine learning, generative models refer to those models that try to represent data distribution or the process of data generation. For content generation, generative adversarial networks (GANs) and diffusion models are popular

since these two models are proven to be effective in image generation, 3D object generation, and 3D scene generation.

GANs, first proposed by Goodfellow et al. [182] in 2014, are designed to learn a mapping from a Gaussian distribution to a data distribution through an adversarial training strategy. Specifically, a generator G and a discriminator D are simultaneously trained. The generator G learns to generate data samples from the latent data distribution, while the discriminator D aims to distinguish between real samples and those generated by G. During training, G aims to minimize the probability of D making correct classifications, effectively pushing the generated samples closer to real data. Recently, GANs have achieved significant success in image generation tasks, prompting researchers to explore their performance in 3D generation tasks. To this end, the generator G is trained to generate 3D data representations such as point clouds, voxels, meshes, or neural implicit representations. Achlioptas et al. [183] proposed two networks, r-GAN and l-GAN, to handle the generation of point clouds. The former directly learns from raw point cloud data, while the latter incorporates a pre-trained autoencoder, achieving promising results. Knyaz et al. [184] presented Z-GAN, which utilizes correspondences between 2D silhouettes and slices of a camera frustum to predict a voxel model of a scene with multiple object instances. For implicit representations, Luo et al. [185] applied adversarial training with spherical mapping to model the implicit surfaces of objects, resulting in smoother and more realistic results. Schwarz et al. [186] proposed a model that combines NeRFs with GANs, enabling the synthesis of high-resolution images.

However, it is challenging for GANs to generate data with extremely complex, high-dimensional distributions. Alternatively, diffusion models derived from the classical score matching method learn the data sampling process to find a path for a Gaussian noise input to reach the real data distribution [187]. These models can outperform GANs after being trained on a large amount of data. The key idea of diffusion models is to transform the original data distribution into a simpler distribution, such as a Gaussian distribution, through a series of noise-driven steps called the forward process. Then, the model learns to reverse this process, known as the inverse process, to generate new samples similar to the original data distribution. Researchers have also combined denoising diffusion models with various types of 3D representations to explore their effectiveness in 3D generation. Luo et al. [188] proposed a diffusion probability model for generating point clouds, modeling the reverse diffusion process for point clouds as a Markov chain conditioned on a certain shape latent. Liu et al. [189] trained a diffusion model to generate deformable tetrahedral grids, and meshes can be extracted from the generated grids. Shap-E [190] and SDF-Diffusion [191] integrate diffusion models with implicit representations. The former directly generates parameters of implicit functions, which can be rendered as textured meshes and NeRFs. The latter generates a low-resolution signed distance function in the first stage and performs high-fidelity super-resolution in the second stage.

6.2 Object generation

Object generation is the basis for building virtual scenes. It involves creating detailed and realistic 3D models of individual objects from scratch or based on limited input data. This process is pivotal in VR applications, where the authenticity and accuracy of generated objects significantly influence the overall experience. Unlike simple 2D image generation, 3D object generation requires a comprehensive understanding of an object's geometry, texture, and material properties.

As shown in Table 5, recent work builds upon diffusion models and large-scale datasets, basically including lifting 2D generative models, imposing multi-view images as priors, and training 3D native generation models.

6.2.1 *Lifting 2D generative models*

To leverage significant advancements in 2D image generation, as demonstrated by recent innovations such as DALL·E [192], Imagen [193], and Stable Diffusion [194], many approaches have adopted image-based techniques, focusing on lifting 2D images into 3D structures or using 2D images as priors. Poole et al. [195] were pioneers in this field, introducing score distillation sampling (SDS) and utilizing 2D image generation with viewpoint prompts to produce 3D shapes through NeRF [30] optimization. Despite the intriguing concept, early attempts often struggled to consistently yield high-quality and diverse results, frequently requiring repeated parameter adjustments and lengthy optimization processes. Subsequent enhancements in SDS have explored the potential of extending the concept to various neural fields [196–201], ranging from DMTet [202] to the latest 3D Gaussian Splatting [35]. Contemporary modifications have significantly improved performance. Seo et al. [203] and Li et al. [204] attempted to add a consistency module to utilize camera and semantic information. Chen et al. [205] introduced normal map supervision to improve the convergence speed. Wang et al. [206] proposed variational score distillation, which infers their distribution using a particle-based variational framework, enhancing the diversity and quality of generated samples. However, 2D image diffusion models used in SDS still lack an explicit understanding of geometry and viewpoint. The absence of perspective information and explicit 3D supervision can result in the so-called multi-head

Table 5 Summary of studies related to 3D object generation.

Classification	Method	Venue	Core technology	3D representation
Lifting 2D generative models	Dreamfusion [195]	ICLR 2023	Score distillation sampling (SDS)	NeRF
	Magic3d [196]	CVPR 2023	DMTet + SDS	Mesh
	Points-to-3d [197]	ACM MM 2023	Depth-guided SDS	NeRF
	HIFA [198]	ICLR 2024	Variance regularization of z-coordinates	NeRF
	Dreamtime [199]	ICLR 2024	Nonincreasing time sampling	NeRF
	Hd-fusion [200]	WACV 2024	multiple noise estimation processes	NeRF
	GSGEN [201]	CVPR 2024	3DGS + SDS	3DGS
	3DFuse [203]	ICLR 2024	3D consistency injection module	NeRF
	Sweetdreamer [204]	ICLR 2024	Aligned geometric priors	NeRF/Mesh
	Fantasia3D [205]	ICCV 2023	Normal-guided SDS	Mesh
	Magic123 [208]	ICLR 2024	Hybrid 2D-3D SDS	NeRF
	Prolificdreamer [206]	NeurIPS 2024	Variational SDS	NeRF
Imposing multiview images as priors	Zero-1-to-3 [207]	ICCV 2023	View-conditioned diffusion	NeRF
	MVDream [212]	ICLR 2024	Multi-view diffusion	NeRF
	USD [209]	arXiv 2023	Unbiased SDS	NeuS
	DreamGaussian [210]	ICLR 2024	3DGS+SDS	3DGS
	DreamCraft3D [211]	ICLR 2024	Bootstrapped diffusion	Mesh
	SyncDreamer [213]	ICLR 2024	3D-aware feature attention	NeuS/NeRF
	Imagedream [214]	arXiv 2023	Image-prompt multi-view diffusion	NeRF
	Zero123++ [215]	arXiv 2023	Enhanced Zero-1-to-3	SDF volumes
	Instant-3D [216]	ISCA 2023	Sparse-view ViT	Implicit fields
	Wonder3D [217]	CVPR 2024	Cross-domain diffusion	Mesh
	Richdreamer [218]	CVPR 2024	Normal-depth diffusion	Hybrid
	One-2-3-45 [221]	NeurIPS 2024	Generalizable NeuS	NeuS
3D Native generation models	CLAY [240]	TOG 2024	Native large-scale 3D diffusion transformer	Mesh
	Polygen [228]	ICML 2020	Autoregressive sequence modelling	Mesh
	MeshGPT [229]	CVPR 2024	GPT-inspired decoder-only transformer	Mesh
	XCube [230]	CVPR 2024	Hierarchical voxel latent diffusion model	Sparse voxel grids
	CraftsMan [241]	arXiv 2024	Normal-based geometry refinement	Mesh
	Direct3D [242]	NeurIPS 2024	Native large-scale image-to-3D generative model	Mesh

Janus problem, where realistic 3D renderings fail to maintain view consistency, and each rendered view is perceived as the front view.

6.2.2 *Imposing multi-view images as priors*

To improve viewpoint consistency, subsequent work introduces viewpoint information to generate consistent multi-view images as prior. Liu et al. [207] introduced an innovative approach by training an additional mapping from the transformation matrix to the pretrained stable diffusion model to incorporate viewpoint information into the image generation process. This strategy allows the network to acquire prior knowledge regarding viewpoint positions and distributions, thereby enhancing the overall generation quality. Alternative solutions also attempt to employ SDS to optimize a coherent neural field [208–211], but they generally require a long optimization time. Shi et al. [212] proposed an enhanced view-aware self-attention mechanism and viewpoint embedding method to directly generate consistent multi-view images. Subsequent studies [213–219] have further improved the consistency of multi-view image generation, enhancing the quality of object generation. Unlike the aforementioned methods that directly use sparse-view neural surface field (NeuS) [220] reconstruction to obtain geometry, Liu et al. [221] had gone one step further in One-2-3-45 to train generalizable NeuS on 3D datasets to achieve better quality. Since the above methods are based on 2D image supervision for generating results, they focus more on the quality of the rendered images and often overlook the geometric fidelity of the generated results. Consequently, the generated geometries tend to be incomplete and lack detail.

6.2.3 *3D native generation models*

To address the challenges of image-based methods, a class of solutions that natively generate 3D objects has emerged.

Large reconstruction model. The large reconstruction model (LRM) treats the generation problem as a single-view or sparse-view reconstruction, utilizing a vision transformer as the backbone network to directly reconstruct implicit representations that include both color and density attributes. One-2-3-45, though is viewed as using 2D image priors, is a pioneer in this class of methods for their clever use of NeuS as a geometry proxy and revealing the possibility of imposing 3D shape priors. Subsequent studies, such as Instant-3D [216], LRM [222,223], DMV3D [224] and TGS [225], have introduced various strategies to enhance generation quality. Xie et al. [226] combined simple shapes to create additional datasets, thereby improving the quantity of the dataset. However, these techniques still focus on minimizing volumetric rendering loss rather than explicitly generating surfaces, which results in rough or noisy geometries.

Explicit geometric representations model. Training directly on 3D datasets and generating explicit geometric representations can clearly address surface quality issues. Various representations have been attempted, including point clouds, meshes, and voxels. Nichol et al. [227] utilized point clouds to achieve a consistent representation of geometry and employ a transformer-based diffusion model for direct denoising on the point clouds. This approach stands out for its simplicity and efficiency; however, it encounters significant challenges in converting the generated point clouds into accurate, standard mesh surfaces. Polygen [228] and MeshGPT [229] take a different approach by natively representing meshes through points and surface sequences. These models are capable of producing extremely high-quality meshes, however, they rely on small, high-quality datasets, which limit their broader applicability. Ren et al. [230] proposed XCube, simplifying geometry into multi-resolution voxels before diffusion. Though this strategy facilitates the process, it still faces challenges in managing complex prompts and supporting a broad range of downstream tasks, limiting its overall flexibility. Methods for generating objects with explicit representations are highly dependent on their respective required datasets, thereby limiting the available data volume. Clearly, a unified representation method that can leverage all types of data is crucial for enhancing the generative capabilities of these models.

Implicit geometric representations model. Introducing signed distance functions (SDF) or occupancy fields and training directly on the processed 3D data is a common solution. Such approaches provide a more explicit mechanism than NeRF for learning and extracting surfaces but require the latent encoding of watertight meshes for generation. Park et al. [231] and Yariv et al. [232] utilized optimization techniques to create unique representations for each geometry in the training dataset. However, the optimization process is sluggish, which hampers training efficiency. Subsequent methods such as SDFusion [233], LAS Diffusion [234], and ShapeGPT [235] leverage intuitive 3D variational autoencoders (VAE) to encode geometry and reconstruct SDF fields, significantly mitigated the issue. Nevertheless, these methods are primarily trained and tested on ShapeNet [236], a dataset with limited richness and quantity, which constrains the diversity and quality of the generated models. Gupta et al. [237] employed a triplane VAE for both encoding and decoding SDF fields. On the other hand, Shap-E [190], 3DShape2VecSet [238], and Michelongelo [239] adopt a different trajectory by utilizing transformers to encode the input point clouds into parameters for the decoding networks, signifying a shift towards more sophisticated neural network architectures in 3D generative models. Furthermore, Zhang et al. [240] proposed an improved data processing method to accommodate datasets of varying sources and quality. They customized a large-scale diffusion transformer compatible with multiple controls to achieve high-quality object generation. By introducing post-processing and physically based rendering (PBR) material generation modules, this method demonstrated potential for integration into industrial pipelines. Li et al. [241] further proposed CraftsMan, which refines the generated results by using 2D normal map diffusion, thereby enriching geometric detail. Wu et al. [242] attempted to use convolutional decoders and triplane representation in the VAE decoder instead of Transformers.

Overall, methods that directly generate objects in implicit fields have demonstrated significant advantages in terms of generation speed, quality, and diversity. However, there is still room for improvement in the quantity and quality of training data, especially when compared to the 2D image datasets used for training stable diffusion. As the community continues to expand training datasets with more diverse 3D graphics and corresponding textual descriptions, we anticipate reaching new levels of quality and complexity in object generation.

6.3 Scene synthesis

Unlike the aforementioned 3D object generation methods, scene synthesis focuses more on perceiving and understanding objects within the scene, thus generating complex scenes with reasonable layouts and imaginative content. Achieving realism and visual coherence in scenes can be challenging due to the intricate interplay of various elements and the overall expression and aesthetic appeal. Recent methods integrate semantic understanding and hierarchical sequential generation into the aforementioned generation model to enhance the overall coherence. The basic methods of scene synthesis include implicit scene synthesis and explicit scene synthesis. Figure 11 shows the

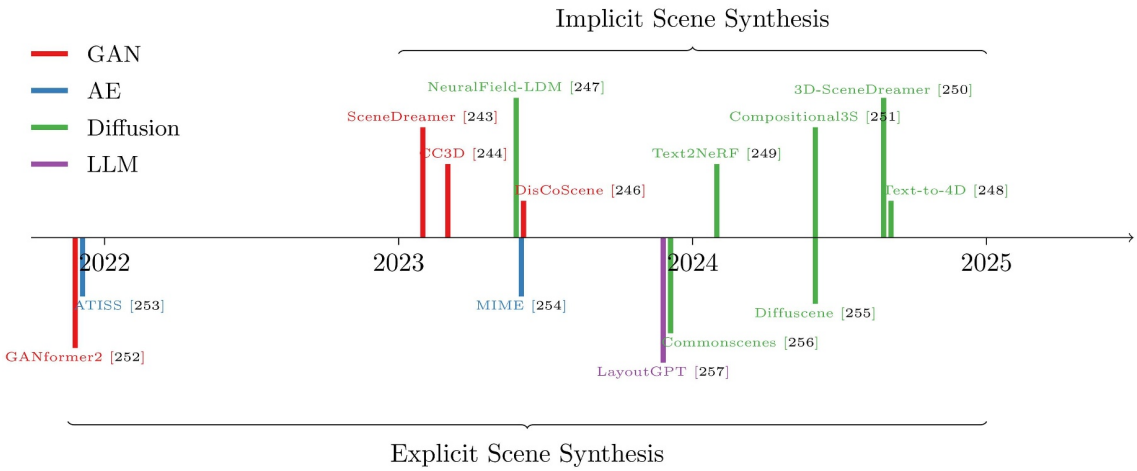

Figure 11 Timeline of the development of scene generation technologies in recent years.

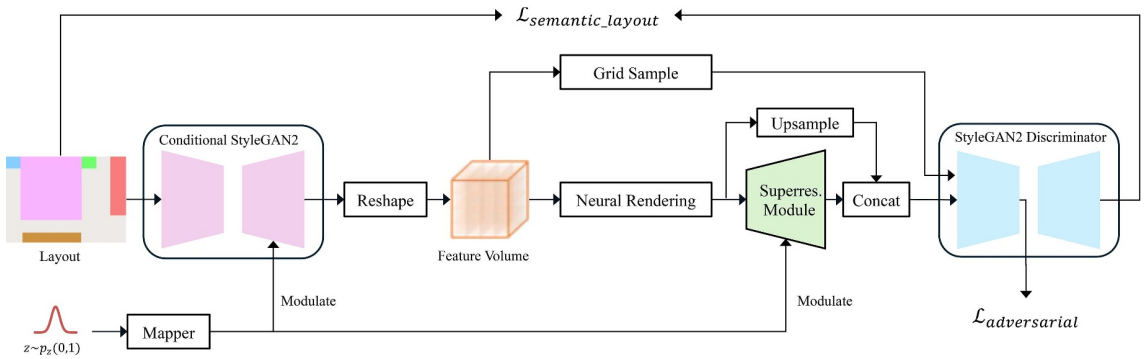

Figure 12 The pipeline of CC3D in the work of Bahman et al. [244]. CC3D takes a floorplan projection of the semantic scene layout and a noise vector as inputs. A 2D feature field is generated by a conditional StyleGAN-V2 backbone based on the given layout, and then the channels are reshaped into a 3D feature volume. This 3D feature volume is queried using trilinear interpolation and subsequently decoded into color and density using a small MLP. A superresolution module is used to upsample volume-rendered images to the target resolution, and a standard StyleGAN-V2 discriminator is used. In order to ensure semantic consistency between the layout and the rendering, equidistant coordinates are sampled from the feature volume, and features are processed with a semantic segmentation decoder added to the discriminator.

development of recent scene synthesis methods in these two aspects, which illustrates the trend of evolution from GAN and AE-based methods to LLM and diffusion-based methods.

6.3.1 *Implicit scene synthesis*

Implicit scene synthesis leverages continuous functional representations to generate scenes. SceneDreamer [243] is an unconditional generative model that achieves unbounded 3D scene generation from in-the-wild 2D image collections only with a BEV (bird's-eye-view) scene representation. To achieve controllable scene generation, some methods introduce constraints like layout, text, and image priors. Bahmani et al. [244] injected 2D layout information as a prior into the 3D scenes generative model, simultaneously achieving performance and efficiency. Figure 12 visualizes the framework of their GAN-based implicit scene synthesis. However, due to its lack of composition modeling, the model fails to generate infinite 3D scenes. To address this issue, BerfScene [245] employs a BEV-conditioned equivariant representation, facilitating seamless composition and infinite-scale scene generation. Xu et al. [246] regarded an abstract object-level representation as the scene layout and proposed a 3D-aware generative model to spatially disentangle the scene into object-centric generative radiance fields. Compared to layout priors, text or image-based controls are a more intuitive solution. Kim et al. [247] designed a scene auto-encoder to encode images into an implicit neural field representation. Zheng et al. [248] proposed a two-stage unified approach for text- and image-driven dynamic 3D scene generation. Zhang et al. [249] provided a pipeline to optimize a NeRF model with text-related scene priors. The scene priors consist of a content prior generated from a text-driven 2D diffusion model and a geometric prior obtained from a monocular depth estimation method. Zhang et al. [250] put forward a text-driven unified solution for both 3D indoor and outdoor scene generation that can also facilitate navigation using arbitrary 6-DOF camera trajectories, benefiting from its tri-plane-based implicit representations. Applying

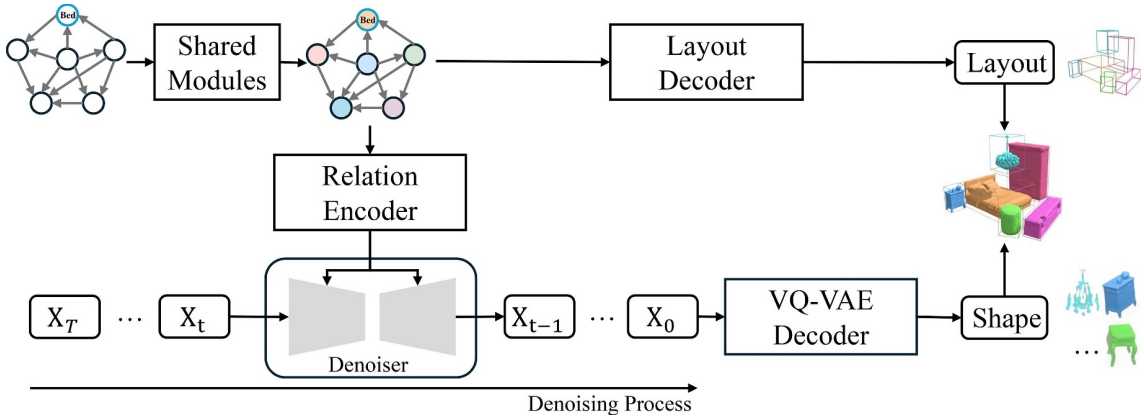

Figure 13 (Color online) The pipeline of CommonScenes in the work of Zhai et al. [256]. CommonScenes consists of shared modules and two collaborative branches layout branch and shape branch. A box-enhanced contextual graph is fed into the contextual encoder, yielding a joint layout-shape distribution. A graph manipulator is then optionally adopted to manipulate the graph for data augmentation. Next, the updated contextual graph is fed into the layout branch and shape branch for layout regression and shape generation, respectively. In the shape branch, the relation encoder is leveraged to encapsulate global scene-object and local object-object relationships into graph nodes, which are then conditioned to the denoiser in LDM via the cross-attention mechanism to generate the shape latent back in some steps. Finally, a frozen shape decoder (VQ-VAE) reconstructs a shape using the latent shape. The final scene is generated by fitting the shape to the layouts.

layout and text guidance simultaneously, Po et al. [251] presented a locally conditioned diffusion for compositional scene diffusion, allowing control over semantic components with text prompts and bounding boxes for seamless transitions.

6.3.2 *Explicit scene synthesis*

Explicit scene synthesis, on the other hand, focuses on the direct construction of scenes from predefined or generated components, such as objects and layouts, offering more control and interpretability. Hudson et al. [252] introduced GANformer2, which abandons traditional black-box GAN architectures, significantly improving the efficiency, controllability, and interpretability of scene layout. ATISS [253] is a novel autoregressive transformer architecture used to create diverse and plausible synthetic indoor environments with only the room type and its floor plan provided. It is trained end-to-end as an autoregressive generative model, using annotated 3D bounding boxes as supervision. Yi et al. [254] proposed mining interaction and movement to infer 3D environments (MIME), which is an indoor scene generation model to produce furniture layouts consistent with human motion. By taking generated objects and human motions in the scene as input, it predicts the next plausible object and generates scenes that are more diverse and believable by incorporating human body information. Tang et al. [255] introduced DiffuScene, a diffusion network to synthesize collections of 3D indoor objects by denoising a set of unordered object attributes. It generates three-dimensional instance attributes stored within the unordered object set and assigns each object the most similar geometric shape retrieved, which concatenates features of different attributes, including position, size, orientation, semantics, and geometric features. Zhai et al. [256] proposed CommonScenes, a fully generative model that can transform scene graphs into corresponding controllable 3D scenes. Its pipeline consists of two branches: one branch predicts the overall scene layout via a variational autoencoder, while the other branch generates compatible shapes through latent diffusion, capturing the relationships between global scene objects and local objects in the scene graph while preserving shape diversity. Figure 13 visualizes the framework of their diffusion-based explicit scene synthesis. Feng et al. [257] investigated how large language models can act as visual planners by generating layouts based on textual conditions, collaborating with visual generation models. They proposed a method called LayoutGPT, which uses a style sheet language to write context visual demonstrations, achieving excellent performance in 3D indoor scene synthesis.

In summary, both implicit and explicit scene synthesis methods have their unique advantages for scene generation tasks. Implicit methods offer a more organic and fluid approach to scene creation, while explicit methods provide greater control and precision. As the field continues to evolve, we can expect to see even more sophisticated techniques that combine the strengths of both approaches to create truly immersive and dynamic virtual scenes.

Table 6 We categorize recent studies on neural physics simulation by the type of physical phenomena they target—rigid bodies, soft bodies, fluids, and others—while analyzing both their geometric representations and neural network architectures.

Classification	Method	Venue	Representation	Model
Rigid-body	Groth et al. [263]	ECCV 2018	Mesh	CNN
	Ehsani et al. [262]	CVPR 2020	Mesh	Encoder-decoder
	Zesch et al. [258]	SIGGRAPH Asia 2023	Mesh	DNN
Soft-body	Lyu et al. [264]	TVCG 2020	Mesh	CNN & GAN
	Santesteban et al. [268]	CVPR 2021	Mesh	VAE & diffused model
	Bertiche et al. [270]	TOG 2022	Mesh	Encoder-decoder
	Wang et al. [265]	CVPR 2023	Mesh	VAE
	Zong et al. [277]	SIGGRAPH Asia 2023	Particle & grid	Neural field
	Romero et al. [278]	SIGGRAPH Asia 2023	Mesh	Neural descriptor fields
	Feng et al. [280]	CVPR 2024	Particle	NeRF
Fluid	Yang et al. [284]	CAVW 2016	Grid	ANN
	Ma et al. [302]	TOG 2018	Grid	RL
	Ummenhofer et al. [293]	ICLR 2019	Particle	Continuous CNN
	Xiao et al. [285]	CGF 2019	Grid	CNN
	Xiao et al. [286]	TVCG 2020	Grid	CNN
	Kim et al. [301]	TOG 2020	Particle	CNN
	Yan et al. [305]	TOG 2020	Mesh & particle	GAN
	Chu et al. [283]	TOG 2021	Grid	GAN
	Guo et al. [291]	TOG 2021	Grid	CNN
	Aurand et al. [292]	TOG 2022	Grid	CNN
	Chu et al. [289]	TOG 2022	Grid	PINN
	Prantl et al. [300]	NeurIPS 2022	Particle	CNN
	Deng et al. [287]	TOG 2023	Grid	InstantNGP-like INR
	Ren et al. [303]	TOG 2022	Particle & grid	RL
Others	Sanchez-Gonzalez et al. [309]	ICML 2020	Particle	GNN
	Li et al. [312]	ICLR 2023	Particle & grid	NeRF
	Zhang et al. [313]	TOG 2020	Particle	RL
	Jiang et al. [314]	SIGGRAPH 2024	Mesh	3DGS

7 Physical simulation

Physical simulation plays a crucial role in enhancing user experience in VR environments by enhancing user immersion, improving user interactivity, and supporting specific professional needs. First, by providing physical animations that adhere to real-world laws, users are more likely to trust the VR content and become more deeply immersed in the virtual experience. Second, the real-time evolution of physical phenomena offers intuitive and physics-based feedback, significantly enhancing the experience with immediate and realistic interactions. Lastly, accurate physics simulation allows VR applications to extend into professional domains such as educational experiments, medical surgeries, and engineering operations, providing a specialized and advanced VR experience. Learning-based techniques have recently introduced a revolutionary methodology to physics simulation. These techniques enhance conventional numerical algorithms for physics simulation, which have been developed over decades, to efficiently generate high-quality results for specific physical phenomena, significantly improving the user experience. From a methodological perspective, we categorize these learning-based techniques into two major areas: neural physics simulation, and differentiable physics simulation. As the dominant branch, neural physics simulation focuses on developing neural network models that replicate, replace, or enhance conventional simulators, supporting tasks such as physics solving, reconstruction, generation, augmentation, and control. We summarize representative research works in Table 6, highlighting the diverse underlying geometric representations and their corresponding neural network architectures. In contrast, differentiable simulation extends traditional numerical simulators by enabling gradient backpropagation through the entire simulation process, facilitating applications in system identification, optimization, control, and inverse problem-solving. As shown in Figure 14, differentiable simulation techniques have been increasingly adopted across various simulation frameworks. In the following sections, we will discuss the progress of research in learning-based methods in physics simulation, categorized by physical phenomena for more intuitive navigation—covering areas such as rigid-body dynamics, soft-body dynamics, fluid dynamics, and other types of physical systems.

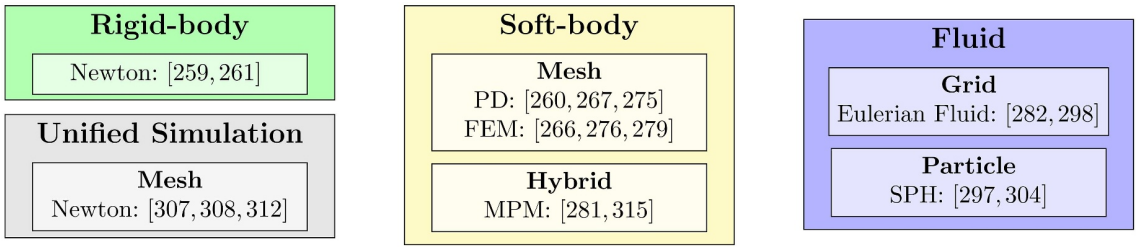

Figure 14 (Color online) We provide a summary of differentiable simulation studies, organized according to their geometric representations and adopted simulation frameworks.

7.1 Rigid-body simulation

Rigid-body simulation mainly focuses on the multi-body system, involving multi-body collision, friction, and various joint constraints within articulated bodies, all of which can be improved through learning-based approaches. Nvidia [258] reformulated collisions as a novel smoothed integral to address sampling issues commonly found in many classic collision-handling algorithms. Since the integral is difficult to calculate numerically, they trained an integrated neural network to represent the collision fields which can efficiently solve collisions for numerous bodies. Their method achieves approximately 100 times speed-up compared to the optimized sampling-based method while exhibiting 2–3 times lower relative error under similar runtime settings. As for differentiable multi-body dynamics, Qiao et al. [259] introduced an efficient method for differentiable simulation of articulated bodies. By deriving the gradients of the contact solver using spatial algebra and the adjoint method, their approach runs 10× faster while consuming 100× less memory footprint compared to traditional autodiff tools. Similarly, in another work [260], they improved projective dynamics with a top-down matrix assembly algorithm and used a new matrix splitting method to apply a generalized dry friction model designed for the soft continuum in their system. Geilinger et al. [261] extended the unified treatment of frictional contact for both rigid and deformable bodies to the domain of robotic motion control. In addition to directly applying differentiable rigid body physics to the simulation of more complex systems, the simulation results of rigid bodies can also be used to infer interactions with rigid objects in video data [262]. This can help neural networks gain a better understanding of the underlying physical principles present in the video. Groth et al. [263] adopted a similar approach, leveraging rigid body simulation to enable models to gain physical intuition. This allows the models to autonomously construct stable stack structures and even restore balance to initially unstable stacks.

7.2 Soft-body simulation

Soft-body simulation is widely used for materials of various dimensions, including hair, cloth, and interactive volumetric soft bodies. By applying neural networks to their fixed topologies, learning-based methods enhance the efficiency and accuracy of dynamics, as well as provide improved interaction, reconstruction, and generation capabilities.

7.2.1 *Hair and cloth simulation*

In virtual reality, the realism of hair and cloth motion is essential for achieving lifelike character fidelity, whether for humans or animals. Recent advancements in learning-based methods have dramatically enhanced the simulation of hair and cloth, leading to a more immersive and engaging VR experience.

For hair simulation, Lyu et al. [264] proposed the first CNN-integrated framework for diverse hairstyles, achieving visually realistic hair simulation at interactive speeds (6.4 FPS with over 60k strands, compared to 0.4 FPS with full simulation). Pre-trained on representative hairstyles, the neural interpolator and fine-scale displacement generator exhibit robust generalization across new hairstyles and poses, making it suitable for interactive VR applications.

Wang et al. [265] introduced a two-stage data-driven approach that models hair independently of the head. The first stage learns hair geometry, tracking, and appearance for state compression, while the second stage samples temporally adjacent hair encodings to train a temporal transfer module for dynamic modeling.

Regarding cloth simulation, Liang et al. [266] differentiated a cloth simulator to estimate material properties and control cloth motion. In their work, the cloth simulation is embedded as a layer within neural network frameworks, providing an effective and robust method for modeling cloth dynamics, self-collisions, and contacts. Li et al. [267] developed a PD-based differentiable cloth simulator that efficiently handles gradients in the presence of complex contact events. To address garment-body collisions, Santesteban et al. [268] proposed a novel data-driven method for

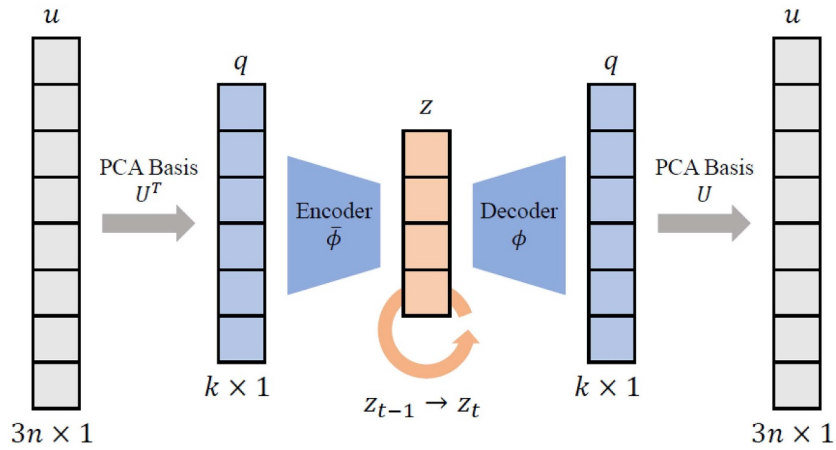

Figure 15 (Color online) A key strategy in learning-based accelerated soft-body simulation in the work of Fulton et al. [274] is to encode dynamic features into a reduced-order latent space.

virtual try-on. They introduced a diffused character model to extrapolate body surface properties, which are then used to project garments to a canonical space. This representation was used to train a generative model with a novel self-supervised collision term, effectively solving garment-body inter-penetrations. Meanwhile, Bertiche et al. [269] proposed an unsupervised deep learning approach to obtain realistic pose space deformations via implicit physics-based simulation. This method is efficient and easily integrated. Furthermore, Bertiche et al. [270] introduced the first method to learn cloth dynamics unsupervised, employing a novel disentangled architecture that enhances generalization for new motion augmentation. Reconstructing the appearance and motion of cloth from real-world data is another important topic for cloth animation. It provides highly customized cloth generation and improves the authenticity in the VR environment. Zhang et al. [271] presented a network model called dynamic neural garments, which takes the target skeleton motion as input and efficiently outputs realistic dynamic garment appearances from a desired viewpoint. To enhance the physical fidelity of cloth animation on virtual characters for VR users, Xiang et al. [272] proposed a physically inspired clothing appearance network that generates realistic cloth simulations on photorealistic characters. Neural cloth animation can be further applied in interactive applications. Wang et al. [273] presented a deep-learning approach for semi-automatic garment animation. Their system enables users to define the desired garment shape in keyframes for interactive editing and visualization. For new character motions, the latent representation automatically generates plausible garment animations at interactive rates.

7.2.2 *Volumetric soft-body simulation*

In addition to the contribution of hair and cloth motion to the physical fidelity of VR environments, volumetric soft-body simulation also plays a significant role. Traditional volumetric soft-body simulations are often inefficient and challenging to control. Neural networks mitigate these issues by learning the dynamics of volumetric soft-body simulations.

Fulton et al. [274] proposed an approach that solves deformable solid dynamics using a variational formulation of implicit integration in the latent space. As shown in Figure 15, this reduced model, trained on example deformations with an autoencoder, accelerates simulation by operating in a low-dimensional nonlinear latent space. To better control volumetric soft-body simulations, Du et al. [275] developed a fast, differentiable simulator based on projective dynamics and a differentiable collision handling algorithm. This simulator accelerates backpropagation by exploiting the prefactorized Cholesky decomposition, allowing physics priors to integrate more effectively into data-driven approaches for dynamical systems involving soft-bodies, resulting in a 4–19 times speedup compared to standard Newton's method. Huang et al. [276] developed a general differentiable solver that integrates the incremental potential contact (IPC) method, supporting both static and dynamic problems while enabling differentiation with respect to geometric and physical parameters. Zong et al. [277] introduced a hybrid framework combining neural networks and physics for modeling elastoplasticity and fracture. They used neural stress, deformation, and affine fields in MPM, achieving a significant reduction of approximately 10 times in computation time and around 100000 times reduction in memory usage. Unlike previous works focused on specific object pairs, Romero et al. [278] developed a neural model that supports general rigid collider shapes with a novel collider descriptor. This learning-based deformation model produces detailed animations and enables object exploration and manipulation.

Applying neural soft-body techniques in reconstructed VR environments enables physically simulating and in-

teracting with virtual objects, enhancing the user's VR experience. Yang et al. [279] adopted the differentiable backward solver proposed by Du et al. [275] to develop an implicit neural representation for controlling active soft bodies. Their method is applicable to volumetric soft bodies and facial expressions, offering potential for interactive applications in virtual reality. Feng et al. [280] proposed PIE-NeRF, integrating physics-based hyperelastic simulations with NeRF to generate realistic elastodynamics of real-world objects. The geometry of the objects is sampled in a meshless manner, followed by spatial model reduction, enabling versatile simulations at interactive rates. Users can interact with the objects by applying external forces in 3D scenes. Zhang et al. [281] proposed PhysDreamer, enabling real-time, physics-based interaction with 3D objects using the differentiable MPM method. They utilized prior knowledge of object dynamics learned from 3D Gaussian pre-trained video generation models to enable static 3D objects to respond dynamically to interactive stimuli in a physically plausible manner.

7.3 Fluid simulation

Fluid simulation, characterized by its complex physical properties and evolving intricate geometry governed by the Navier-Stokes equations, presents significant challenges in the physical simulation. Learning-based techniques open up new possibilities for understanding fluid materials, geometry, and dynamics and for further reconstructing, enhancing, and generating fluid phenomena.

7.3.1 *Gas simulation*

Gaseous phenomena such as wind, smoke, and fire are common in the real world. Researchers have long studied these basic fluid phenomena and are now integrating them with learning-based techniques to create more realistic or faster simulations in various applications. The relevant studies are generally divided into neural dynamics simulation and neural effect generation. We will commence with the neural dynamics simulation of gaseous phenomena.

Takahashi et al. [282] introduced a differentiable smoke simulator integrated with neural networks for learning dynamics and solving control problems, enabling efficient gradient computation and one-way fluid-solid coupling with sub-grid details, outperforming prior techniques for fluid control and inverse problems. Chu et al. [283] introduced a data-driven adversarial model for deriving fluid velocity fields from density maps, enabling control via obstacles, parameters, energy, and vorticity for more interpretable and controllable fluid simulations.

Researchers are also exploring the neural physics solver, a novel technique that utilizes neural networks to predict physics dynamics efficiently. By encoding and learning the physics state in latent space, neural solvers achieve superior performance and accuracy compared with traditional numerical solvers. A data-driven projection methodology, pioneered by Yang et al. [284], harnesses the power of artificial neural networks to expedite the projection phase in smoke simulations, ensuring consistent computational efficiency regardless of the complexity of the scene. Xiao et al. [285] proposed a learning-based flow correction method using a deep convolutional neural network to quickly preview Eulerian smoke simulations, accurately matching low-resolution simulations to high-resolution counterparts. They [286] also presented a neural solver that leverages a deep convolutional neural network to efficiently tackle large-scale Poisson systems in Eulerian smoke simulations, achieving a significant speedup of up to two orders of magnitude in the projection step while maintaining both accuracy and versatility. Deng et al. [287] introduced neural flow maps, a neural solver that leverages implicit neural representations and flow map theory to achieve state-of-the-art vortical smoke simulations using spatially sparse neural fields, preserving intricate vortical structures with high fidelity. Compared with modern numerical solvers, it reduces advection error by about an order of magnitude.

While neural physics solvers aim to replicate physical behavior, other methods focus on reconstructing simulations from limited data and augmenting them with stylistic control. Eckert et al. [288] introduced ScalarFlow, a large-scale dataset of real-world smoke reconstructions, and a framework for accurate physics-based recovery from sparse videos, highlighting complex buoyancy-driven flows for graphics, vision, and learning applications. Chu et al. [289] presented a novel approach for high-fidelity fluid reconstruction from sparse videos, leveraging Navier-Stokes physics and neural networks, resolving fluid-obstacle interactions and ambiguities to enable robust reconstructions.

Researchers have been investigating gas style transfer, a method for controlling the appearance of generated gas in a unique way. It involves adding customizable details without violating the physical laws of smoke flow. Kim et al. [290] introduced a neural style transfer method for smoke simulations, enabling content-aware manipulation with natural image features. Guo et al. [291] proposed a neural network approach for volumetric style transfer, efficiently creating heterogeneous single-scattering albedo volumes from 2D style images and enabling diverse translucent effects for 3D models. Aurand et al. [292] introduced an improved volumetric neural style transfer method for smoke simulations, enabling faster, simpler, and more controllable stylizations while eliminating camera-dependent artifacts through a feed-forward neural network.

7.3.2 *Liquid simulation*

Unlike gases, liquid phenomena—such as water, oil, or honey—are typically more intricate, often involving highly dynamic free surfaces and strict incompressibility. In recent years, learning-based methods have emerged as a novel approach for simulating a range of complex liquid behaviors.

In the realm of neural dynamics solver, Ummenhofer et al. [293] presented a Lagrangian liquid simulation utilizing convolutional networks with spatial convolutions on dynamic particles, surpassing previous methods in both accuracy and speed for various material simulations. Shao et al. [294] introduced TIE, a transformer-based approach for particle-based liquid simulations, which captures particle interactions without relying on explicit edges, demonstrating superior performance and generalization compared to graph neural network methods. Li et al. [295] developed MPMNet, a hybrid data-driven framework that integrates the material point method (MPM) with neural networks to achieve efficient and precise liquid-solid interactions, maintaining physical accuracy while enabling numerical acceleration (28× speed-up compared with the conventional method).

Reconstruction of liquid effects is often challenging due to the transparent nature of liquids. As a result, current efforts primarily focus on inferring the internal dynamics of liquids from observable fluctuations on the liquid surface. Franz et al. [296] introduced a novel volumetric flow reconstruction method employing global transport formulation and learned self-supervision, enabling realistic fluid motion reconstruction from sparse views. Guan et al. [297] presented NeuroFluid, an unsupervised two-stage network designed for liquid dynamics reconstruction from visual observations. This method utilizes a particle-driven renderer and a transition model to estimate fluid physics accurately. Xiao et al. [298] proposed a novel framework for reconstructing liquid dynamics from sparse observations. By leveraging a differentiable simulator and divergence-free eigenfunctions, this method achieves physically consistent and efficient fluid reconstruction.

Another set of works aims to enhance neural liquid simulation with finer detail, strict adherence to physical constraints, or specific artistic styles. Roy et al. [299] introduced a deep up-scaling technique for high-resolution liquid details, leveraging neighborhood convolutions and particle-based interpolation. In addition, Prantl et al. [300] introduced a method using antisymmetrical convolutions to strictly conserve momentum in learned liquid simulations, resulting in enhanced details. Besides, a neural style transfer approach for 3D fluids [301] was presented, leveraging a Lagrangian particle representation to enhance artistic control over liquid details while improving temporal consistency and reducing computational time.

Controlling or interacting with liquids is a topic that creates immersive and engaging experiences for users, especially in contexts highly relevant to VR applications. For liquid control, considering the complexity of the liquid system, recent researchers prefer to apply reinforcement learning techniques to achieve the indirect control of fluid-solid coupling systems. Ma et al. [302] proposed a neural-network-driven controller, trained by reinforcement learning, to control liquid jets and rigid body interactions at simulation boundaries, generating plausible animations for 2D tasks. Ren et al. [303] proposed an adaptive learning-based controller for coupled fluid-solid systems, excelling in liquid control tasks through meta-reinforcement learning and a novel task representation. Additionally, a differentiable SPH-based fluid-rigid coupling simulator [304] enables efficient rigid body control in liquid environments, addressing gradient issues and minimizing computational costs. For interacting with liquids, Yan et al. [305] presented a novel system that enables amateur users to generate realistic liquid splashes in minutes, leveraging a conditional GAN trained on physics-based simulations and considering stroke trajectory and speed for intuitive liquid interaction in a VR environment. Feng et al. [306] demonstrated the integration of physics-based animations of solids and fluids with 3D Gaussian Splatting to create realistic scenes, focusing on liquid interactions through enhanced kernel normals and physically based rendering for dynamic surface reflections.

7.4 Other physical simulation

In addition to these specific fields of physics simulation, other studies aim to develop general neural frames or techniques for multi-physics simulation. Hu et al. introduced diffTaichi [307,308], a framework based on the Taichi language, enhancing both the efficiency and productivity of general-purpose differentiable physics simulations. Sanchez-Gonzalez et al. [309] proposed graph network-based simulators (GNS) to reproduce various particle-based physics simulations, including fluids, rigid, and deformable. By treating underlying particles as graph nodes, GNS encodes the input state in a latent graph and predicts the dynamics via learned message-passing on the graph. Nvidia has implemented a multi-physics field simulation framework called SimNet [310], which takes spatial location and physical parameters as inputs and learns the PDE solutions through a fully connected network. SimNet has been applied in both forward simulations involving turbulent details and complex obstacles and inverse problems like industrial design optimization. They validate SimNet by conducting a comparative study with OpenFOAM and a commercial solver, where SimNet achieves 45000× and 135000× acceleration, respectively, while maintaining a

15% prediction error compared to OpenFOAM's 4.5%. Recently, Wang et al. [311] advanced the grid-based neural simulation by employing a multi-resolution hash grid. To improve the accuracy and flexibility in elastic deformation and vortical flows, they introduced two key enhancements: a high-order differential operator for optimization efficiency and an octree-based neural geometry sampling method to accelerate interface searching.

Leveraging these multiphysics simulation techniques, researchers are pioneering new frontiers in VR-related applications, such as real-world physical reconstruction, immersive physical interactions, and physical content creation. Building upon MPM, Li et al. [312] proposed physics augmented continuum neural radiance fields (PAC-NeRF) to reconstruct the physical effects from multi-view videos, capturing both physical parameters and geometries. PAC-NeRF ensures the physical plausibility of reconstruction by integrating NeRF techniques with continuum mechanics through a hybrid Eulerian-Lagrangian representation and an MPM differentiable simulator. As for the interaction, Zhang et al. [313] introduced a reinforcement learning framework with a position-based dynamics (PBD) simulator to train characters in manipulating amorphous materials. Jiang et al. [314] implemented VR-GS, a highly realistic interactive system in VR based on the Gaussian Splatting techniques. Starting from a Gaussian Splatting scene, the system constructs a simulation-ready environment through segmentation, inpainting, and mesh reconstruction and incorporates a PBD simulator for an immersive interactive VR experience. Huang et al. [315] introduced DiffVL, a method allowing non-expert users to communicate soft-body manipulation tasks using vision and natural language. Their method leverages large language models to assist the differentiable physics solver in handling long-horizon, multistage tasks. Physically based content creation is an emerging topic. Qiu et al. [316] proposed feature splatting to enable language-driven, physics-based Gaussian Splatting scene editing. Using multi-view photos and text prompts as inputs, feature splatting constructs a Gaussian Splatting segmented scene with a large-scale 2D vision model, automatically assigning physical attributes to scene components. An MPM-based physics engine is further integrated to create realistic physics animation within the Gaussian Splatting scene.

8 Virtual character

The research on virtual characters is mainly concerned with the digitization and animation of real-world characters or generated characters. The two most commonly used types of virtual characters in VR applications are avatars and agents. Avatars are usually motion-driven, and currently, text and audio inputs are used to generate motions, based on which animatable avatars are explored to create realistic and adaptable virtual characters. Research on agents has focused on exploring agents that can autonomously respond to the environment, thus enhancing the interactivity and realism of the virtual experience.

8.1 Motion generation

Motion generation aims to generate realistic and varied human movement sequences, which have various applications in the VR domain, such as teaching and training, and contextual extrapolation. The primary challenge of motion generation lies in creating sequences that are both perceptually realistic and diverse. Based on different input types, the generation of gestures is mainly divided into text-based gesture generation, including action commands and general text-based methods, and audio-based gesture generation, including speech and music-based methods. In Table 7, we list some representative research studies and the datasets they used [317, 331, 334, 335, 338–349].

8.1.1 *Text-based motion generation*

Text-based motion generation uses textual descriptions to create corresponding motion sequences. The core concept behind it is combining natural language processing with motion generation, enabling computers to understand textual descriptions and produce corresponding actions. It mainly includes two types: generating motion sequences from specific action commands (action-to-motion) and generating motion sequences from general textual descriptions (text-to-motion).

Action-to-motion aims to produce precise and accurate motion sequences based on explicit action instructions with high controllability and predictability such as "walk", "jump", or "turn left". Action2Motion [317] introduces a novel VAE framework based on Lie Algebra to generate diverse and natural sets of human motions for specified action categories. Petrovich et al. [318] introduced ACTOR, a transformer-based architecture with positional encodings to generate variable-length motion sequences directly, avoiding mean pose regression and introducing the sequence-level embedding, unlike frame-level approaches such as Action2Motion. SA-GCN [319] combines the self-attention mechanism with graph convolutional networks (GCN), dynamically focusing on key past frames to capture structural information in action sequences effectively. Likewise, Kinetic-GAN [320] combines the advantages of GANs and

Table 7 Summary of studies related to human motion generation.

Classification	Method	Venue	Model	Representation	Dataset
Action	Action2Motion [317]	ACM MM 2020	VAE	Rotation	Action2Motion [317], NTU-RGB+D 120 [338]
	ACTOR [318]	ICCV 2021	VAE	Rotation	Action2Motion [317], NTU-RGB+D 120 [338], UESTC [339]
	SA-GCN [319]	ECCV 2020	GAN	Keypoints	NTU-RGB+D 120 [338], Human3.6M [340]
	Kinetic-GAN [320]	WACV 2022	GAN	Keypoints	NTU-RGB+D 120 [338], Human3.6M [340], NTU-RGB+D [341]
	ODMO [321]	MM 2024	VAE	Keypoints	Action2Motion [317], UESTC [339]
	PoseGPT [322]	ECCV 2022	VAE	Rotation	Action2Motion [317], BABEL [342], GRAB [343]
	MultiAct [323]	AAAI 2023	VAE	Keypoints/rotation	BABEL [342]
General text	Text2Action [324]	ICRA 2018	GAN	Keypoints	MSR-VTT [344]
	JL2P [325]	3DV 2019	Regression	Keypoints	KIT Motion Language [345]
	TEMOS [326]	ECCV 2022	VAE	Keypoints	KIT Motion Language [345]
	TM2T [327]	ECCV 2022	VAE	Keypoints	HumanML3D [346], KIT Motion Language [345]
	AvatarCLIP [328]	TOG 2022	VAE	Rotation	AMASS [347]
	MotionCLIP [329]	ECCV 2022	Regression	Rotation	BABEL [342]
	MoFusion [330]	CVPR 2023	Diffusion	Keypoints	HumanML3D [346], BABEL [342], AI choreographer [348]
Speech	Speech2Gesture [331]	CVPR 2019	GAN	Keypoints	Speech2Gesture [331]
	Audio2Gestures [332]	ICCV 2021	VAE	Keypoints/rotation	Speech2Gesture [331], Trinity [349]
	Yoon et al. [333]	TOG 2020	GAN	Keypoints	Human3.6M [340]
Music	ChoreoMaster [334]	TOG2021	Motion Graph	Rotation	ChoreoMaster [334]
	Pc-dance [335]	MM 2022	Motion Graph	Rotation	Pc-dance [335]
	MNET [336]	CVPR 2022	GAN	Rotation	AI Choreographer [348]
	EDGE [337]	CVPR 2023	Diffusion	Rotation	AI Choreographer [348]

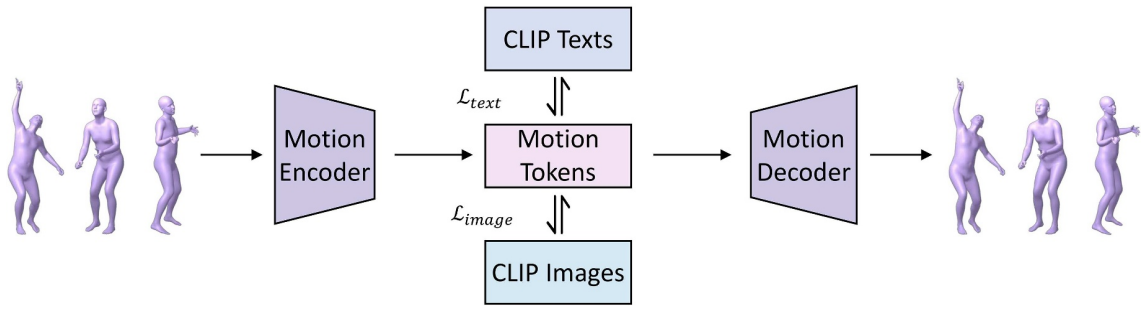

Figure 16 (Color online) MotionCLIP in the work of Tevet et al. [329] trains a motion auto-encoder to simultaneously reconstruct motion sequences while aligning their latent representations with corresponding text and image representations in CLIP space.

GCNs to generate human motion sequences from the latent space directly. Furthermore, Lu et al. [321] proposed ODMO for generating 3D human motion sequences solely based on action types, utilizing contrastive learning for effective style discovery and hierarchical trajectory control in its encoder-decoder architecture. PoseGPT [322] compresses human motion into a discrete latent space based on an auto-regressive transformer, allowing generation without relying on observed past motions. Additionally, some approaches focus on generating motions involving multiple actions. For instance, MultiAct [323] achieves this by using a unified recurrent generation system to produce realistic long-term 3D human motion sequences from multiple action labels.

The shortcomings of action-to-action in creating lifelike and adaptable movements are compensated by the fact that text-to-action can generate motion sequences based on natural language inputs, thus allowing a wider range of expressive motions. Text2Action [324] is a GAN-based SEQ2SEQ model with a text encoder converting input sentences into feature vectors, enabling an attention-based action decoder to generate corresponding actions from the encoded text sequences. Ahuja et al. [325] proposed the joint language-to-pose model (JL2P), which learns a joint embedding space of these two modalities and employs a curriculum training strategy to handle sequences of varying complexity. In recent years, the variational autoencoder has drawn a surge of interest because of its capability to learn complex latent representations and generate diverse and realistic outputs. TEMOS [326] utilizes VAE and incorporates a text encoder to generate expressive body motions. Similarly, TM2T [327] uses both motion and text tokens, integrates the motion2text module into the inverse alignment process of the text2motion training pipeline, and uses VAE to train them. Moreover, some models utilize the vision-language model to complete the text-to-motion task. AvatarCLIP [328] leverages the contrastive language-image pre-training (CLIP) to supervise neural human generation, enabling zero-shot generation of novel animated avatars. MotionCLIP [329] aligns 3D motion with text labels in CLIP-space using a transformer-based auto-encoder, enhancing semantic understanding and motion generation through CLIP's visual insights, as shown in Figure 16. Additionally, diffusion models are employed for their ability to generate high-quality and detailed outputs. Dabral et al. [330] introduced MoFusion, a denoising-diffusion-based framework featuring a 1D U-Net for faster reverse diffusion, capable of generating motion sequences from both music and text.

8.1.2 *Audio-based motion generation*

Auditory signals are much more complex than text, and their features contain spectrograms, mel-frequency cepstral coefficients (MFCCs), chroma features, and so on. When it comes to motion generation through audio, two aspects are usually taken into account, namely the semantic information and the temporal structure, i.e., the rhythm, the beat, and the timing information of the notes. Thus, this section focuses on two subtasks: speech-to-gesture and music-to-dance correspondingly.

The goal of the speech-to-gesture task is to generate corresponding hand gestures based on speech input, making the virtual characters perform natural and appropriate gestures while speaking. Some methods utilize only text, while more rely solely on audio, and some combine both modalities. Ginosar et al. [331] generated motion from speech by mapping audio to pose, with an adversarial discriminator ensuring realistic gestures. Audio2Gestures [332] splits the latent code into shared and motion-specific codes to generate diverse gestures with random sampling and relaxed motion loss. Yoon et al. [333] took the text, audio, and speaker identity all into consideration and generated gestures under a multimodal context.

The music-to-dance task aims to generate motion sequences that correspond to a given piece of music, improving the overall aesthetic and expressive quality of the dance movements for virtual characters. Some methods are based on the classical motion graph framework, while some are based on GANs. On the one hand, ChoreoMaster [334] captures music-dance connections through choreomusical embeddings and integrates them into a graph-based synthesis framework, and generates a dance sequence to synchronize with the input music, reflecting its style, rhythm, and structure. Pc-dance [335] utilizes a music-to-dance alignment embedding network and pseudo-label of the rhythm, generating dance with a posture constraint. On the other hand, MNET [336] integrates a music style code into the generator and designs a multi-task discriminator for per-style classification. EDGE [337] uses a transformer-based diffusion model paired with a music feature extractor to generate editable dance motions from music.

8.2 Avatar creation

3D avatar creation involves the reconstruction of the human body and the creation of animatable avatars. While reconstructing the human body focuses on basic human digitization techniques to recover body geometry and texture information for further avatar creation, creating animatable avatars emphasizes the development of avatar models that can be driven by novel motions and rendered with generative appearances for lifelike animation and multi-modality manipulation.

8.2.1 *Human reconstruction*

Human reconstruction aims to create a digital 3D model of a human from various forms of input data, including images, videos, or depth scans. Techniques for human reconstruction can be divided into traditional and modern reconstruction based on the input data type and methods used.

Traditional reconstruction methods usually rely on complex hardware devices to capture observation data of the target human body, such as multi-view images and depth information. Based on the data collected by human capture systems consisting of multiple arranged industrial cameras and depth cameras, researchers employ the structure from motion (SfM) [350] technique to determine camera poses, use the multi-view stereo (MVS) [351,352] method to calculate depth information and exploit the depth fusion algorithms [353–356] to construct the 3D mesh model. Schönberger et al. [357] proposed a per-pixel camera view selection strategy based on the PatchMatch [358] framework in the MVS stage to enhance dense reconstruction, achieving more accurate reconstruction of fine structures and weak texture areas such as human hair and skin.

Though achieving good performance, the high computational complexity of traditional pipelines renders them unsuitable for real-time interactive applications. Therefore, modern methods utilize depth maps acquired by depth sensors for real-time reconstruction through direct depth fusion. DynamicFusion [359] first achieves dynamic human body reconstruction from a single depth camera, but it encounters difficulties when handling fast movements and matching depth maps with the previous frame's model. DoubleFusion [360] uses the skinned multi-person linear model (SMPL) [361] to capture human motions and use it as a regularization term to constrain the optimization process of the deformation graph, preventing it from getting stuck in local optima. However, since the SMPL model does not encompass clothing dynamics, its optimization of the deformation graph for loose clothing remains constrained. Motion2Fusion [362] uses high-speed depth sensors to reduce the motion amplitude between frames, enabling more accurate deformation graph recovery. RobustFusion [363] introduces a data-driven model reconstruction algorithm, achieving high-quality dynamic human reconstruction using a monocular RGB-D camera.

Table 8 Summary of studies related to animatable avatar.

Classification	Method	Venue	Representation	Input data
Human avatar	Neural Body [368]	CVPR 2021	NeRF	Sparse-view videos
	MetaAvatar [369]	NeurIPS 2021	Neurl SDF	Monocular depth images
	Neural-GIF [370]	ICCV 2021	SDF	3D scans
	LEAP [371]	CVPR 2021	Occupancy field	3D pose
	SCANimate [372]	CVPR 2021	Implicit function	3D scans
	SNARF [373]	ICCV 2021	Implicit surface	3D scans
	Bagautdinov et al. [374]	TOG 2021	CVAE	3D pose
	Animatable NeRF [375]	ICCV 2021	NeRF	Multi-view videos
	AvatarRex [376]	TOG 2023	NeRF	Multi-view video
	HumanNeRF [377]	CVPR 2022	NeRF	Monocular video
	Neural Actor [378]	TOG 2021	NeRF	3D pose
	Animatable Gaussian [379]	CVPR 2024	3DGS	Multi-view images
Head/facial avatar	Jackson et al. [380]	ICCV 2017	Voxel grid	Single image
	Feng et al. [381]	ECCV 2018	2D UV position map	Single image
	Jiang et al. [382]	IEEE TIP 2018	3D morphable model	Multi-view images
	Wu et al. [383]	CVPR 2019	3D morphable model	Multi-view images
	Bai et al. [384]	CVPR 2020	3D morphable model	Multi-view images
	Yenamandra et al. [385]	CVPR 2021	Implicit 3D morphable model	3D scans
	Wang et al. [386]	IEEE TMM 2021	SDF	Multi-view images
	Zheng et al. [387]	CVPR 2022	Implicit deformation field	Monocular videos
	Zhuang et al. [388]	ECCV 2022	NeRF	Single image
	Hong et al. [389]	CVPR 2022	NeRF	Single image
	AD-NeRF [390]	ICCV 2021	NeRF	Monocular videos (including audio)
	Gao et al. [391]	TOG 2022	NeRF	Monocular videos
Avatar with generative appearance	ClipFace [398]	SIGGRAPH 2023	3D morphable model	Text
	Rodin [399]	CVPR 2023	NeRF, diffusion model	Text/image
	DreamHuman [400]	NeurIPS 2024	NeRF	Text
	imGHUM [401]	ICCV 2021	SDF	3D mesh
	HumanNorm [402]	CVPR 2024	Diffusion model	Text

Another way for human reconstruction is model-free, addressing the loose clothing limitation by predicting occupancy values of a volumetric space. Pixel-aligned implicit function (PIFu) [364] aligns pixel-level local features to the overall object context through fully convolutional operations and utilizes an MLP to infer depth and occupancy values directly from these features. However, depth ambiguities from single-image inputs limit PIFu's ability to handle complex poses, and the quality of 3D datasets also restricts the effectiveness of PIFu-based methods. The NeRF series [30, 91, 92] of studies offers users an easy way to create high-quality, renderable human models. Users simply need to have the target person remain still and capture a set of images using a monocular camera from various angles. These images can then be used to optimize the NeRF of the human body. For example, DoubleField [365] combines the merits of both surface field and radiance field for high-fidelity human reconstruction and rendering. Still, achieving real-time and high-quality reconstruction with NeRF under sparse view conditions remains challenging. DiffuStereo [366] integrates the continuity of diffusion models with current learning-based iterative stereo to achieve high-quality human depth estimation.

8.2.2 *Animatable avatar*

Human Reconstruction focuses on how to recover the 3D geometric information of the human body from 2D image observations, achieving dimensionality upscaling. In contrast, the goal of animatable avatar modeling is not only to obtain a 3D human reconstruction consistent with the image but also to enable new motion driving and generate corresponding images. The driveability of the human body and head/facial models is crucial for various VR applications containing digital humans, including immersive virtual meetings and virtual companions that necessitate reconstructed digital humans controlled by human postures. The representative works are listed in Table 8.

Human avatar. The SMPL and SMPL-X [367] models achieve driveability by the pose parameters θ. Based on this, researchers have applied deformation techniques to obtain deformable 3D human models. Neural Body [368] combines the SMLP model with NeRFs, integrating observations from different video frames using human priors to reconstruct free-viewpoint videos from sparse-view videos. However, this method is unable to produce dynamic details accompanying pose changes, such as clothing wrinkles, leading researchers to explore the use of implicit fields. MetaAvatar [369] and Neural-GIF [370] map every point in space to a standard pose, such as the T-pose, using inverse skinning methods, and then learn the non-rigid deformations of clothing in the standard pose. LEAP [371] and SCANimate [372] use neural networks to learn forward and inverse skinning fields and impose cycle-consistency constraints between them. To better adapt to new poses, SNARF [373] proposes a differentiable forward skinning model that uses an iterative root-finding method to query any point's corresponding position in the standard pose within the pose space.

To achieve textured results, traditional methods typically begin by reconstructing a polygonal mesh model of a

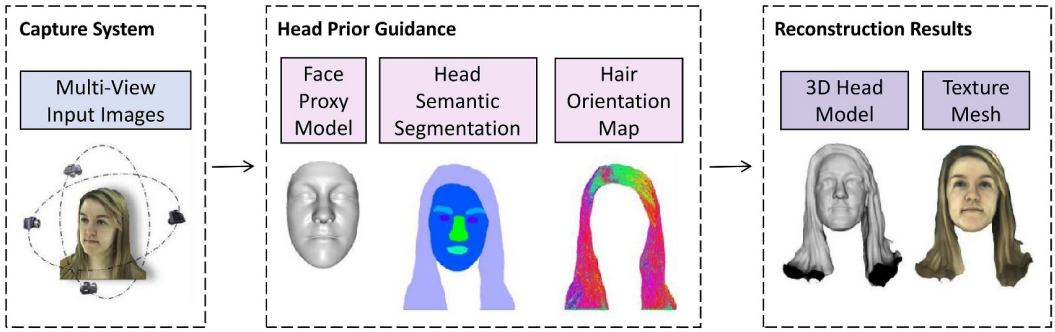

Figure 17 (Color online) Wang et al. [386] employed implicit differentiable rendering methods to model head geometry and incorporated priors to enhance reconstruction accuracy and robustness.

specific object with textures and materials and then generating its motion, resulting in generally lower image quality. Bagautdinov et al. [374] from Meta Reality Lab modeled dynamic geometry and appearance by decomposing the driving signals, achieving the first high-fidelity human avatar modeling, which provides the possibility of obtaining more realistic avatar images.

In recent years, neural volume rendering techniques have demonstrated advantages in rendering highly realistic free-viewpoint effects for both static scenes and dynamic sequences. Some approaches use NeRF and deformation fields to depict dynamic human bodies. Animatable NeRF [375] combines human pose parameters with skin weight fields to form deformation fields, driving models based on skeletal skinning. AvatarRex [376] learns NeRF-based full-body avatars from video data, providing expressive control over the body, hands, and face while supporting real-time animation and rendering. HumanNeRF [377] decomposes dynamic scenes into standard static scene models and deformation fields, integrating information from different times explicitly into static scene models through deformation fields. This method reconstructs 3D dynamic human body models from sparse-view videos, surpassing Neural Body. These methods decompose the dynamic human body into a deformation field based on inverse skinning and a neural radiance field in a standard pose. By mapping the NeRFs of different poses into the standard space, better pose generalization is achieved. However, these methods have limited capability in synthesizing appearance details, making it difficult to generate realistic dynamic wrinkle changes. Therefore, Neural Actor [378] proposes feature texture maps as additional input signals, encoding high-frequency appearance details onto 2D texture maps, thereby reducing the learning burden of NeRF networks. Moreover, Animatable Gaussian [379] uses 2D CNNs and 3D Gaussian Splatting to create high-fidelity avatars, learning a parametric template from input videos that adapt to loose clothing like dresses.

Head/facial avatar. The modeling of head avatars plays a crucial role in enhancing the realism of digital humans, as the face provides rich personal information such as race, age, gender, emotions, personality traits, and physical condition. Recent advancements have focused on improving the realism and accuracy of facial features.

Head avatar modeling based on explicit mesh representation utilizes parameterized facial models and various types of inputs to construct facial models. Input data includes single images, multi-view images, and RGB videos. The key to the 3D reconstruction from a single image is to establish correspondence from 2D-pixel points to 3D spatial points. Jackson et al. [380] employed volumetric representation to model 3D facial models, designing a convolutional neural network to directly regress 3D facial meshes from single facial images. Feng et al. [381] introduced a 2D representation method named UV position mapping to record complete facial 3D positional coordinates. Jiang et al. [382] combined bilinear facial models with shape-from-shading [26], proposing a three-stage hierarchical processing approach to enhance facial detail expression capabilities. Multi-view 3D facial reconstruction improves avatar accuracy; for instance, Wu et al. [383] and Bai et al. [384] incorporated geometric or appearance consistency between multi-views, though such methods often require extensive 3D geometric data for training, with template models performing poorly in hair reconstruction.

Head avatar modeling based on implicit geometry representation takes advantage of the flexibility and variability of this representation method. Yenamandra et al. [385] first proposed a deep implicit 3D deformable model of the complete head, employing signed distance fields to model head shapes, not only constructing front-facing geometric structures, textures, and expressions representing identity but also modeling the entire head including hair. Wang et al. [386] utilized signed distance field geometric representations, introducing facial priors, head semantic segmentation information, and 2D hair direction maps for guided reconstruction, as shown in Figure 17. Zheng et al. [387] employed implicit neural networks to learn 3D head geometry, representing deformations related to expressions and poses through learnable blended shapes and skin weights.

Head avatar modeling based on NeRFs generates high-fidelity portraits and expands application scenarios. Zhuang et al. [388] introduced a NeRF-based head parameterization model MoFANeRF, achieving separate control over appearance, shape, and expression, but the model cannot accommodate hairstyle modeling. Hong et al. [389] proposed a generalized parameterized head model HeadNeRF, semantically decoupling the latent space of head models according to attributes such as expression, shape, and lighting, enabling control over generated attributes and some extent, facilitating the generation of different hairstyles. Guo et al. [390] addressed speech-driven NeRFs with ADNeRF, using extracted speech features as conditional input to achieve cross-modal driving effects. Gao et al. [391] further proposed a personalized semantic facial model, decomposing continuously varying head models into disentangled low-dimensional spaces and semantically informative bases, drawing realistic head images under given expression coefficients and viewing directions, enhancing facial retargeting and expression editing applications. Some approaches generate videos of someone speaking based on input voice, including generating 2D speaking person videos (StyleHEAT [392], Everything's Talkin [393]), generating 3D digital avatar head geometry animations (VOCA [394], CodeTalker [395]), and using NeRFs for voice-driven avatar heads (GeneFace [396], RAD-NeRF [397]), among others.

Avatar with generative appearance. Creating realistic and detailed avatars with generative appearance is another hot topic. For head avatar appearance, some methods use the CLIP for text parsing and utilize diffusion models for supervised image generation or directly generate representations on three planes, such as ClipFace [398] and Rodin [399]. For full-body avatar appearance, DreamHuman [400] introduces imGHUM [401] as a human prior capable of handling loose garments with unique topologies, optimizes a NeRF network using fractionally distilled sampling strategies, and focuses on rendering and optimizing each semantic part of the human body for enhanced texture generation details. HumanNorm [402] is a novel approach for high-quality and realistic 3D human generation by enhancing the model's 2D perception of 3D geometry through normal-adapted and normal-aligned diffusion models.

8.3 Autonomous agent

The agent is an entity that is placed in an environment and senses different parameters that are used to make a decision based on the goal of the entity. The entity performs the necessary action on the environment based on this decision [27]. Agents possess sociability, allowing them to share knowledge and request information from others, and autonomy, enabling independent decision-making and action. They also demonstrate proactivity by using their history, sensed data, and information from other agents to predict future actions and take effective measures to achieve their goals. In recent years, based on techniques such as reinforcement learning and large language model (LLM), research related to agents has made significant progress in two main directions: crowd simulation and autonomous agent.

8.3.1 *Crowd simulation*

Crowd simulation seeks to accurately replicate the motion dynamics of virtual agents, which plays an important role in building realistic and believable virtual environments. Traditional simulation methods can be divided into two categories: microscopic models and macroscopic models. The former focuses on the low-level behavioral details and individual characteristics in the crowd, while the latter treats the crowd as a whole without considering the interaction between agents [28].

In recent years, researchers have adopted data-driven approaches, including using data to calibrate the parameters of the model and extracting behavioral paradigms from the data to improve the realism and stability of the simulation while avoiding the cost of extensive manual modeling [18]. Earlier studies used neural network classifiers to cluster the input states and select feasible behaviors within the clusters, greatly reducing the search space and time overhead of traditional methods [403]. Wang et al. [404] used the neuro-evolution method for simulation, which does not need to use real data to train the network, but it is difficult to ensure the realism of the simulation. Wei et al. [405] proposed for the first time the direct use of neural networks to learn potential movement rules from real data, where they extracted state-action pairs from the data and fed them into the neural network training. Unlike the previous methods, Lee et al. [406] applied the reinforcement learning approach to crowd simulation by designing simple reward functions to generate optimal policies without the need to adjust complex parameters for each scenario, which is more generalizable. Further, Hu et al. [407] introduced the use of control parameters such as preferred speed as input to the policy to generate heterogeneous behaviours for crowd simulations. Ref. [408] introduced configurable crowd profiles that allow real-time control of agent parameters without the need to retrain the learned model. Charalambous et al. [409] used double deep Q-learning's reinforcement learning algorithm to learn reward functions from input data and find optimal strategies for individuals during training with better generalization

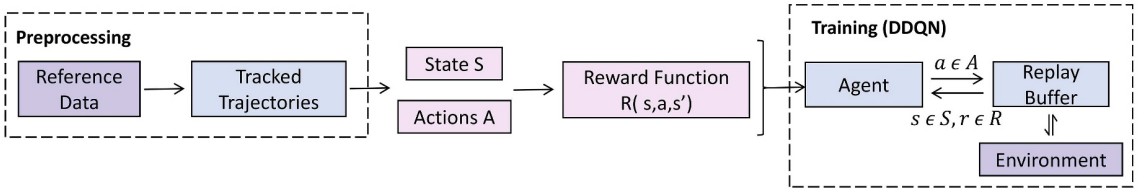

Figure 18 (Color online) The pipeline in the work of Charalambous et al. [409].

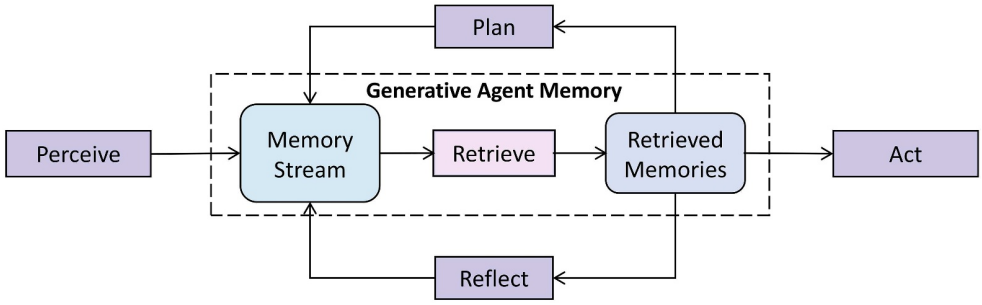

Figure 19 (Color online) The architecture in the work of Park et al. [411].

properties and performance optimization. Their pipeline is shown in Figure 18. These methods greatly enhance the diversity of crowd behaviors and the heterogeneity of virtual crowds.

8.3.2 *Autonomous characters*

Trustworthy behavior of agents can enhance the experience of virtual interaction and has a wide range of applications in various areas such as education and healthcare [410]. Unlike the physical behavioral simulation of intelligent agents, there is also a need to endow them with sociality and autonomy. Early characters were mainly given behavioral rules or scripts by designers, but this obviously does not allow for true autonomy and makes it difficult to construct the complex processes by which they interact with each other. On the other hand, learning-based approaches use a policy function to drive the agent but are mostly applicable to restricted environments, unlike the open world in which humans live [19].

With the remarkable success of the large language model, LLM-based intelligences have more comprehensive knowledge and are able to make informed autonomous behaviors despite the absence of context-specific data. The pipeline of related research is mainly divided into four modules: profile, memory, planning, and action [19]. E.g., in Figure 19, Ref. [411] utilized the powerful prompting function of ChatGPT to propose generative agents that are able to memorize, retrieve, reflect, and interact with other agents, as well as plan and act through a dynamically evolving environment. In general, the profile module reflects the basic psychological and summary information of the character, which can be assigned manually [411,412], generated by LLM [413], and so on. The memory module is designed to resemble the human mind and is divided into short-term and long-term memory. Refs. [414–416] applied different hybrid long- and short-term memory architectures. The planning module empowers the character to make reasonable and reliable decisions, and the action module translates the decisions into concrete behavioral expressions [19]. This shows that the autonomous role design in generation based on AI technology has a wide range of application prospects but also needs to consider moral and ethical issues [410].

To further enhance the emotional intelligence of autonomous agents in interactions, existing AI technologies integrate deep learning and cognitive modeling to enable artificial systems to perceive, express, and adapt to human emotions. Within the framework of affective computing, the core components include emotion perception, emotion-aware expression, and agent emotion modeling [29], which work together to allow AI to naturally adapt to users' emotional states during interactions. Affective computing employs data-driven approaches and theory-driven approaches. Data-driven methods [417] primarily rely on deep learning models such as CNN, LSTM, and GRU for emotion recognition and generation, while theory-driven methods leverage psychological and neuroscience theories to provide AI with more interpretable emotional models. Furthermore, multimodal affective computing combines various perception channels, such as text, speech, facial expressions, and gestures, to improve the accuracy and naturalness of emotion recognition [418].

Moreover, some techniques have been introduced to this field to make an agent's behavior more aligned with interaction requirements and enhance its explainability. Chain of thought (CoT) was originally developed in the

field of natural language processing [419], and it employs a step-by-step reasoning approach to break down complex problems into intermediate steps, thereby improving transparency and accuracy in the reasoning process. A recent study [420] proposed the visualization of thought (VoT) prompting method, which induced spatial reasoning in LLMs by visualizing the reasoning trajectory, thus guiding the subsequent reasoning steps. The utilization of visualization techniques can also help humans better understand the reasoning logic of AI, thus increasing its transparency and credibility.

Recent research increasingly focuses on enabling agents to take proactive roles in interactions. Notable studies include proactive agent [421], which enhances agents' ability to autonomously anticipate user needs and proactively provide services by leveraging environment perception, user need prediction, and autonomous task execution. This is achieved through constructing the ProactiveBench dataset, fine-tuning large language models, and training reward models, ultimately improving interaction fluency and user experience. Similarly, proactive conversational agents [422] enhance proactive interaction by anticipating user needs, actively guiding conversations, and optimizing interaction pacing, resulting in more natural, efficient, and user-aligned conversational experiences.

9 Interaction

Human-computer interaction in virtual reality consists of interactions between the user and virtual objects, avatars, and environments, as well as the user's subjective perceptions. In recent years, much research has focused on using AI techniques to enhance these interactions in various ways. Specifically, this includes three main directions. The first is user behavior recognition in virtual reality environments since accurate behavior recognition is essential to enable virtual reality systems to respond appropriately to user behavior. Second is interaction optimization, which focuses primarily on improving user actions and behaviors in virtual space. Finally, perception analysis and augmentation focus on analyzing the user's perceptions during interactions to enhance the system's ability to provide a more immersive and responsive experience, ultimately improving user engagement and satisfaction.

9.1 Behavior recognition

Users exhibit intentional behaviors through their hands, eyes, and facial. These behaviors and the intent behind them are often complex and diverse. Accurately recognizing them can significantly improve the efficiency and precision of human-computer interactions. In a virtual environment, we have an opportunity to thoroughly monitor and analyze the inner meaning of human behavior in a controlled environment, which makes it especially well-suited for data-driven AI algorithms. We will review the work for recognizing and tracking users' behaviors and identify the underlying intentions that drive them. In Table 9, we compare the algorithms and hardware devices used across different areas of behavior recognition.

9.1.1 *Tracking and recognition*

The purpose of tracking and recognizing user behavior in VR is to accurately and quickly identify interactive inputs so that the correct feedback can be provided in a timely manner to enhance the overall user experience. This tracking and recognition typically focuses on three key aspects: hand, eye, and facial. Mainstream algorithms leverage neural networks, such as CNNs, to estimate the poses of these three aspects. Additionally, methods like SVM and RF are employed for behavior regression and classification, ensuring precise and responsive interactions in the virtual environment.

Hand tracking and recognition. Hand tracking and recognition play a crucial role in enhancing human-computer interaction. However, due to the diverse range of hand movements and the complexity of interaction environments, accurately tracking and recognizing hand movements remains a significant challenge. Several existing approaches focus on improving the model's ability to handle occlusion using neural networks. Mueller et al. [423] introduced a method to estimate hand pose using egocentric RGB-D cameras. Their approach involves two sequentially applied CNNs that estimate the 2D position of the hand center and then regress 3D locations of hand joints. Wang et al. [424] proposed a real-time method using a multi-task CNN for capturing both the skeletal pose and 3D surface geometry of hands. By regressing multiple complementary pieces of information, their multi-task CNN effectively addresses depth ambiguities in RGB data and accurately estimates pose and shape parameters for both hands, as depicted in Figure 20. Han et al. [425] was the first to introduce a real-time hand-tracking system using CNNs on four fisheye monochrome cameras aimed at enhancing VR experiences. They [426] later expanded their work by developing a unified end-to-end framework for multi-view, multi-frame hand tracking that directly predicts 3D hand poses in world space on a VR headset. This framework includes a 3D feature extractor module

Table 9 Summary of studies related to behavior recognition.

Classification	Method	Venue	Core technology	Hardware device
Hand tracking and recognition	Mueller et al. [423]	ICCV 2017	Supervised learning	RGB-D camera
	Rgb2hands [424]	TOG 2020	Supervised learning	RGB-D camera
	MEgATrack [425]	TOG 2020	Supervised learning	Fisheye camera
	UmeTrack [426]	SIGGRAPH Asia 2022	Supervised learning	RGB-D camera
	HOOV [427]	CHI 2023	Supervised learning	Headset, wrist-worn band
	Diliberti et al. [429]	MM 2019	Supervised learning	Motion capture gloves
	Arimatsu et al. [430]	CHI 2020	Supervised learning	Controller with capacitive proximity sensors
	GestOnHMD [431]	TVCG 2021	Supervised learning	Stereo microphones
	Jiang et al. [428]	Virtual Reality 2018	Supervised learning	Force myography, Leap Motion
Eye tracking and recognition	Lu et al. [432]	ISMAR 2020	Supervised learning	Eyeglasses with multiple RGB-D cameras and LEDs
	Wang et al. [433]	ISMAR 2021	Unsupervised learning	near-infrared camera
	Gaze from origin [434]	AAAI 2024	Unsupervised learning	RGB-D camera
	UVAGaze [435]	AAAI 2024	Unsupervised learning	Dual cameras
	DGaze [436]	TVCG 2020	Supervised learning	Headset
	SGaze [437]	TVCG 2019	Optimization	Headset
	Stubbemann et al. [438]	ETRA 2021	Supervised learning	Camera of headset
Facial tracking and recognition	Chen et al. [440]	CVPR 2021	Supervised learning	Phone
	Teng et al. [441]	VRCAI 2016	Supervised learning	Headset with RGB-D camera
Human-object interaction	Where2Act [442]	ICCV 2021	Self-supervised learning	Simulation
	AdaAfford [443]	ECCV 2022	Supervised learning	Simulation
	LEMON [444]	CVPR 2024	Semi-supervised learning	RGB-D camera
	EgoChoir [445]	NeurIPS 2024	Semi-supervised learning	Headset
	EgoLM [446]	arXiv	Supervised learning	Eyeglasses, motion sensors
	Sun et al. [447]	MM 2021	Semi-supervised learning	Multiple cameras
	Wang et al. [448]	CVPR 2024	Supervised learning	Simulation
Viewport analysis	Feng et al. [449]	AIVR 2019	Supervised learning	Headset
	LiveDeep [450]	IEEE VR 2020	Supervised learning	Headset
	Heyse et al. [451]	IEEE VR 2019	Reinforcement learning	Headset
	Instant Reality [452]	TVCG 2022	Supervised learning	Headset, traditional screen
Attention prediction	Delvigne et al. [454]	AIVR 2020	Supervised learning	Headset, EEG
	Li et al. [455]	Virtual Reality 2021	Supervised learning	Headset

that processes single-view or multi-view data to generate 3D features via a feature transform layer and a pose regression module for producing 3D hand pose information.

Some research has utilized information provided by hardware devices to aid neural networks in classification and recognition tasks. Streli et al. [427] introduced HOOV, a hand out-of-view tracking method that leverages multimodal information, including the 3D orientation of the wrist and head pose, as inputs to an RNN and transformer-based model for hand position prediction. This approach enables VR users to interact with objects outside their field of view. Similarly, Jiang et al. [428] combined force myography and leap motion signals as multimodal inputs, employing SVM, decision trees, and neural networks to recognize hand-grasping actions in virtual reality environments. Diliberti et al. [429] applied CNNs to 3D rotation data of finger joints obtained from motion capture gloves to classify user gestures and intents. Arimatsu et al. [430] designed a finger-tracking controller with capacitive proximity sensors to recognize hand movements and developed a dual CNN architecture to estimate hand pose. Chen et al. [431] proposed a gesture-based interaction technique utilizing stereo microphones in mobile phones to detect gestures on VR headsets, employing CNNs for gesture detection and recognition.

Eye tracking and recognition. Eyes enable users to explore and interact with the virtual world, making eye tracking and recognition essential for accurately displaying virtual content and supporting interactive behaviors that require high precision. Lu et al. [432] introduced an eye-tracking solution based on high-order Purkinje reflection images, using an end-to-end CNN to map the characteristics of these images to the vergence and accommodation of the eyes. Recognizing that critical information is often concentrated in edge areas, Wang et al. [433] developed an edge extraction network, leveraging edge generation networks and adversarial learning to predict edge maps. These maps are then fed into an edge-guided segmentation and fitting network based on an MLP to accurately segment and fit ellipses. Additionally, they [434] implemented a gaze estimation network and a gaze frontalization module to rotate the eyeballs toward the front camera, enhancing gaze estimation performance. Furthermore, they proposed an unsupervised gaze estimation framework, adapting the classic single-view estimator based on neural networks to work with dual cameras [435]. Some research focuses on predicting gaze from eye-tracking data. Hu et al. [436] introduced DGaze, a multimodal CNN-based model that integrates object position sequences, head velocity sequences, and saliency features to predict users gaze positions within the viewport of a head-mounted display. Their model achieved an average angular distance error of 7.57°. To tackle the challenge of limited data, Stubbemann et al. [438] employed cycle-GAN for augmenting eye-tracking images and introduced a new training dataset as input for CNNs to predict volumes of interest.

Facial tracking and recognition. Facial tracking and recognition in VR encompasses both face tracking and reconstruction, as well as the recognition of facial expressions with users wearing a head-mounted display. Neural networks are commonly employed in the tracking and recognition of VR users' faces. In the early stages of research,

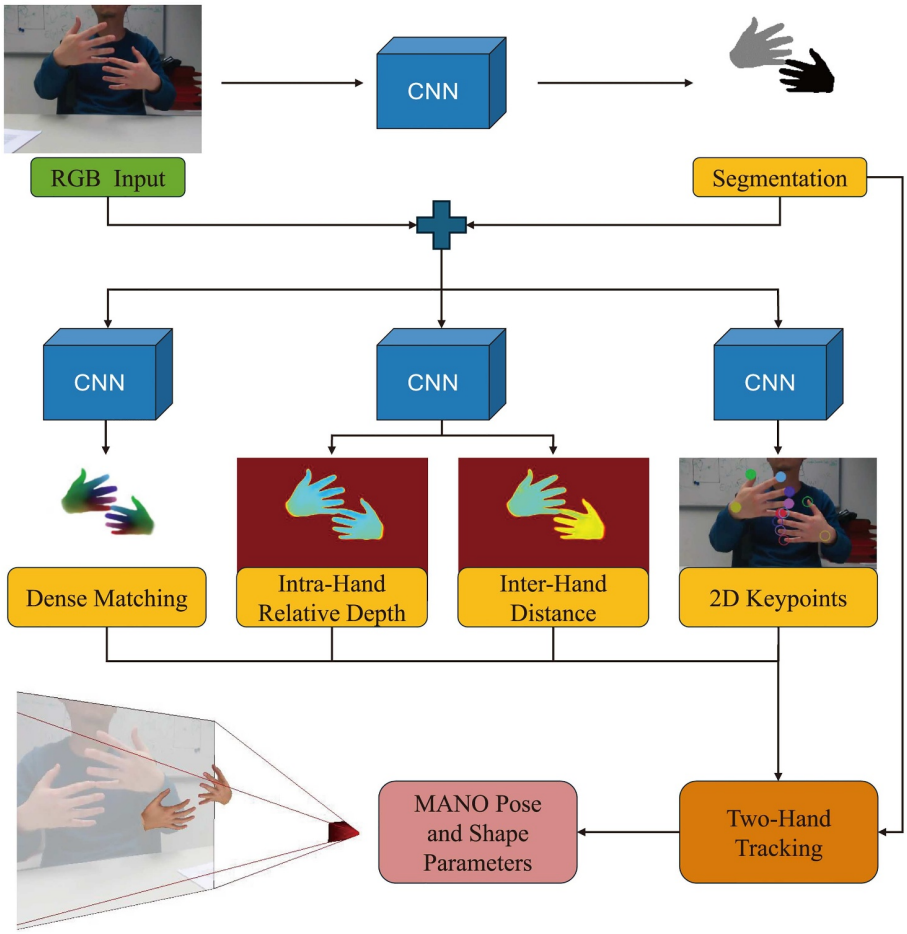

Figure 20 (Color online) RGB2Hands model in the work of Wang et al. [424]. The RGB input image is processed by neural predictors that estimate segmentation, dense matching, intra-hand relative depth, inter-hand distances, as well as 2D keypoints. This is then used within our two-hand tracking energy minimization framework. The outputs are pose and shape parameters of the 3D MANO model of both hands.

Girado et al. [439] introduced a video-based, real-time, low-latency, high-precision 3D facial tracker. This system utilized a central camera and artificial neural networks to pinpoint 2D facial positions and to identify and track upright, tilted, frontal, and non-frontal faces in visually cluttered environments. Building on this, Chen et al. [440] applied deep learning illumination models based on VAE and MLP to recover accurate texture and geometric details from images captured in the wild. They combined these with advanced 3D facial tracking algorithms to enable subtle and robust facial motion transfer, effectively transforming ordinary videos into highly realistic 3D avatars. Facial expression recognition in VR presents an ongoing challenge, particularly because the upper half of the face is often obscured by an HMD. To address this, Teng et al. [441] processed facial data by segmenting the mouth area and then trained a CNN using this processed mouth information to detect expressions.

9.1.2 *Intention recognition*

Human behavior in VR often reflects specific underlying intentions. By extracting and analyzing these intentions, designers can tailor experiences to meet users' personalized needs, especially when working with VR devices that have limited performance capabilities. Current research in this area primarily focuses on user viewport analysis and attention prediction.

Human-object interaction. Human-object interaction involves understanding interactions within scenes, the objects present, and the underlying intentions driving these interactions. Some research has specifically focused on human-centric interaction understanding. These studies primarily leverage deep learning to train models for mapping affordance [442, 443], though many of these models lack contextual consideration of human interactions. Yang et al. [444] exploited the correlation between the interaction counterparts. They used an interaction intention excavation module, curvature-guided geometric correlation module, and contact-aware spatial relation module based on cross attention and transformer to jointly anticipate human contact, object affordance, and human-object spatial

relation. Recently, with the rise of embodied AI, researchers have begun to consider using egocentric information as input. Yang et al. [445] employed modality-wise encoders to extract features from egocentric video, head motion, and object information. These features were then used in a parallel cross-attention framework to uncover interaction concepts and infer the subject's intentions. Incorporating multi-modal sensors, Hong et al. [446] combined sparse motion sensors and egocentric video inputs to prompt LLM for ego-motion tracking and understanding. Other studies have focused on generating virtual human activities based on interaction understanding. Sun et al. [447] proposed an interaction-aware human-object capture framework that integrates occlusion-aware implicit human reconstruction, human-aware object tracking, and neural blending to generate human activities from novel perspectives. Wang et al. [448] leveraged scene affordance as an intermediary to link 3D scene grounding with conditional motion generation, enabling the synthesis of human motion guided by language.

Viewport analysis. Scenes and videos in VR are highly data-intensive and are often streamed in real-time, posing significant challenges for bandwidth management. Researchers frequently analyze the user's viewport to prioritize the transmission of relevant areas, thereby optimizing the delivery process. Feng et al. [449] investigated the use of CNNs to predict user viewports during live streaming, leveraging the dynamic propagation characteristics of VR content. Their experimental results indicated that this method could reduce bandwidth usage by 57%. Building on this, they proposed LiveDeep [450], a hybrid model that combines CNN and LSTM to create an online viewport prediction system. Heyse et al. [451] adopted a two-stage reinforcement learning approach using contextual bandits, which involved movement detection and direction prediction to predict the user's viewport in 360-degree video. Similarly, Chen et al. [452] employed neural networks to estimate the perceptual importance in the 2D image space based on the user's gaze behavior, such as where the user gazes and how the gaze is moved. Then, they mapped this importance to 3D object space for optimizing rendering.

Attention prediction. Attention is a limited scarce resource that users deploy to navigate complex scenes and extract valuable information. Generally, attention analysis in VR environments uses data from eye-tracking and head position sensors embedded in head-mounted displays. Khokhar et al. [453] utilized features collected from VR headset sensors, including angular velocity, positional velocity, pupil diameter, and eye openness, to train a CNN-LSTM classifier capable of detecting distracting objects. Expanding on the traditional focus on head position and eye movement, Delvigne et al. [454] incorporated EEG signals into their model inputs and used a combination of CNN, SVM, RF, and MLP to estimate attention scores. Li et al. [455] further enhanced attention prediction by accounting for the spatial and temporal characteristics of user behavior, integrating contextual information through a hybrid LSTM-CNN model. Fathy et al. [456] argued that lighting features significantly influence visual attention and perception, thus incorporating luminance and contrast data into an ensemble bagged tree model to estimate user attention levels.

9.2 Interaction optimization

Interaction, which includes both motion and manipulation, is a critical aspect of VR. The main challenge in optimizing interactions is to improve naturalness and efficiency within the constraints of available hardware. Research on interaction optimization mainly involves the directions of motion retargeting and manipulation assistance.

9.2.1 *Motion redirection*

The limitations of physical space and hardware in VR can restrict the user's movement and tactile experience, breaking their immersion in the virtual world. Motion redirection seeks to overcome these limitations by exploiting the dominance of visual perception to create the illusion of physical feedback. This is achieved through techniques that subtly manipulate the user's movements, effectively expanding the perceived physical space. The two main categories of motion redirection are redirected walking and haptic retargeting.

Redirected walking. Redirected walking technology allows users to explore vast virtual spaces without the need for large areas of physical space. Reinforcement learning, with its capacity for continuous learning and strategy adaptation, is often employed to handle the complexity of dynamic scenes in redirected walking. Strauss et al. [457] utilized reinforcement learning to train a deep neural network that directly prescribes rotation, translation, and curvature gains, transforming the virtual environment based on the user's position and orientation within the tracked space. Similarly, Chen et al. [458] employed reinforcement learning with a novel dense reward function to jointly consider physical boundary avoidance and consistency of user-object positioning between virtual and physical spaces (Figure 21). They [459] further extended their method for virtual-physical environmental alignment at multiple transferable target positions in passive haptic tasks with novel reward function designs. Lee et al. [460] presented a novel control algorithm called steer-to-optimal-target (S2OT) for real-time planning in redirected walking, designing and training a machine learning model using reinforcement learning and deep Q-learning to estimate optimal steering

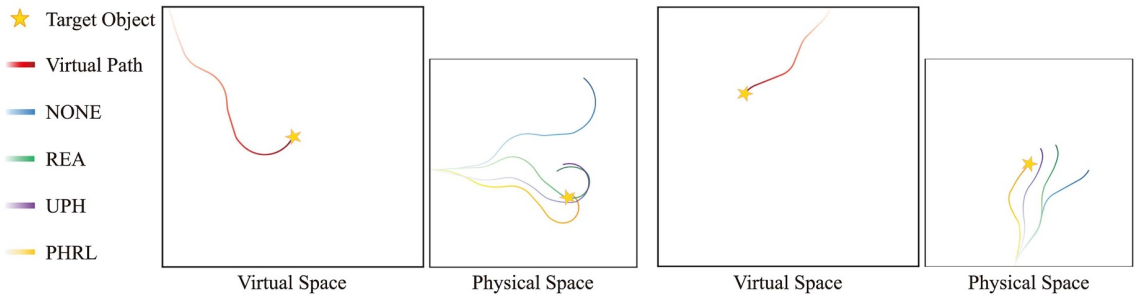

Figure 21 The comparison of different redirected walking methods, which include no redirection (NONE), redirection with ubiquitous passive haptics (UPH), redirection with reactive environmental alignment (REA), and passive haptics with reinforcement learning (PHRL) in the work of Chen et al. [458].

targets. Subsequently, they extended the S2OT approach to multi-user application scenarios. Jeon et al. [461] also explored multi-user scenarios and proposed an optimal space partitioning (OSP) method that dynamically divided the shared physical space in real-time using deep reinforcement learning to allocate optimal sub-spaces to each user and provided optimal steering. Azmandian et al. [462] introduced a static planning strategy called combinatorially optimized pre-planned exploration redirector (COPPER), which optimizes gain parameters for a predetermined virtual path. They conducted a simulation-based experiment demonstrating how adaptation rules could be determined empirically using machine learning. Cho et al. [463] proposed a data-driven path prediction model using LSTM networks trained on user path data collected from a maze-like environment.

Haptic retargeting. Haptic retargeting maps multiple virtual objects onto a few physical props, providing passive haptic feedback by subtly adjusting the virtual environment to align with the real world. Initial approaches focused on predicting user interactions and providing corresponding haptic feedback based on predefined scripts. To extend haptic feedback to interactions with remote objects, the FlyingHand method [464] integrated multimodal perception. It utilized CNNs for object classification based on images captured by drones, providing virtual hand-haptic feedback that aligns with the objects. Yixian et al. [465] introduced ZoomWalls, which uses SVM to predict the surface the user is about to touch based on their direction and walking speed. Clarence et al. [466] enhanced the flexibility of interactive object selection by developing unscripted retargeting. They trained LSTM networks on users' hand-reach trajectories to predict their intended targets. Salvato et al. [467] tackled the issue of sense-to-actuation latency in haptic feedback by introducing a self-attention-based network. This network used a time series of tracked hand poses and virtual object geometry to predict when a user would begin interacting with a virtual object through touch.

9.2.2 *Manipulation assistance*

Manipulation assistance aims to enhance the efficiency and precision of user interactions in VR. The core idea is to leverage contextual information to improve interaction accuracy and intelligence. This includes contextual assistance and selection optimization.

Contextual assistance. Providing contextual assistance in virtual environments can significantly enhance user capabilities and improve their overall experience. For instance, Ge et al. [468] developed a tool for precise assistance during Deep Anterior Lamellar Keratoplasty, which accurately tracks corneal contours using SVM for data annotation and a combination of U-net and CNN for tracking. However, excessive assistance can disrupt the user's experience, reducing immersion. To address this, Alghofaili et al. [469] developed an adaptive navigation assistance system that leverages multimodal information. They utilized a dataset containing user gaze data and orally requested information to train an LSTM network. This network classifies gaze sequences to determine when assistance is required, providing support only when necessary. Seeliger et al. [470] introduced context-adaptive visual cues to minimize visual noise caused by assistance in industrial environments. They employed a deep neural network to decide whether to display assistance based on context related to the user and task, such as task progress, task duration, and the user's head movement.

Selection optimization. Selection in VR is a fundamental task, particularly in exploratory and visual analysis of point clouds and scatter plots. However, VR often struggles with accurate selection due to issues like occlusion and data similarity. Chen et al. [471] addressed these challenges with LassoNet, a hierarchical neural network that learns the mapping between point clouds, viewpoints, and lassos to facilitate the accurate selection of 3D point clouds. Cordeil et al. [472] proposed an interactive machine learning framework to tackle the 3D point cloud selection problem. This framework uses human-in-the-loop classification dialogue to iteratively classify point clouds,

leveraging PCEDNet [473] to enhance selection accuracy.

9.3 Perception analysis and enhancement

Perception is central to shaping the user experience and interaction within virtual environments, as it directly influences not only the degree of immersion but also overall user satisfaction. By analyzing various aspects of perception, such as emotional responses and susceptibility to motion sickness, developers can fine-tune virtual experiences to minimize discomfort and maximize engagement. Furthermore, enhancing perception through advanced content design and personalized experiences allows for the creation of more dynamic and adaptive virtual worlds tailored to meet the specific needs and preferences of individual users.

9.3.1 *Perception analysis*

Perception analysis utilizes physiological indicators to assess emotions and predict motion sickness. The core of this process lies in processing information and creating regression models of user perception through machine learning and deep learning. The primary directions of focus include affective computing and motion sickness prediction.

Affective computing. Human behavior is deeply influenced by emotions, which play a crucial role in how individuals perceive and react to different situations. Affective computing aims to model and respond to these emotional states. Some research has utilized multimodal signals collected from various physiological sources, such as the brain, heart, and skin. For example, Marín-Morales et al. [474] fused multimodal signals, including electroencephalography (EEG) and electrocardiography (ECG), to train an SVM for predicting users' valence and arousal. Building on this, Gupta et al. [475] expanded the approach by incorporating additional input signals and regression models to improve classification accuracy. Their system, AffectivelyVR, identifies personalized emotions in real-time within virtual environments by training various regression models, including SVM, KNN, RF, and Gaussian Naive Bayes, using off-the-shelf EEG and galvanic skin response (GSR) sensors. Some studies also combine affective computing with specific application domains. For instance, Šalkevicius et al. [476] developed a framework for predicting anxiety levels during virtual reality exposure therapy (VRET) using wristband sensors to process GSR, blood volume pulse (BVP) and skin temperature signals. These signals are fed into an RF model, and the extracted features are classified using SVM to determine the user's anxiety level. Vaitheeshwari et al. [477] analyzed soldiers' stress levels by simulating real battlefield conditions in VR. They employed LSTM (long short-term memory) and CNN with inputs from ECG, GST, and eye-tracking signals, assisting SVM and RF models in regressing stress levels.

Motion sickness prediction. Motion sickness, a common discomfort experienced by users of HMD, limits the immersive potential of VR. Despite its prevalence, the mechanisms behind motion sickness are not yet fully understood. Martin et al. [478] proposed a machine learning model to non-invasively detect vertigo in real-time. They trained SVM, Gradient Boosting, and RF models using sensor data and user-reported responses to predict motion sickness, with RF achieving the best results. Liu et al. [479] developed a framework for predicting motion sickness scores, utilizing CNN, ECA, and LSTM models to automatically analyze EEG signals and predict sickness levels.

9.3.2 *Perception enhancement*

Perception enhancement focuses on improving user perception through rich content and personalized experiences. By incorporating AI algorithms, virtual environments can become more intelligent and user-specific. Perception enhancement involves two directions: interactive content design and personalized experience.

Interactive content design. Interactive content design involves creating and optimizing interactive objects, virtual avatars, and narratives within the VR environment. The design of VR content significantly influences users' sense of perception. To enable non-player characters (NPCs) to respond intelligently to participant's interactions, Dobre et al. [480] collected data on user interactions with NPCs and trained a reinforcement learning algorithm, PPO [481], enhanced with LSTM for temporal memory to improve accuracy. Zhao et al. [482] combined LLM with a virtual city, allowing participants to interact with LLM-driven characters in VR to simulate real-life language-use scenarios, which reduced barriers to learning and improved learning efficiency. Alghofaili et al. [483] optimized VR design by focusing on computational attention. They proposed a novel data-driven optimization method that predicts gaze duration using decision trees, RF, and SVM, and utilizes a Markov chain Monte Carlo technique to optimize the placement of elements within the virtual environment, maximizing user attention.

Personalized experience. VR holds significant potential for creating personalized experiences tailored to the unique needs of specific populations, such as children with disabilities or athletes undergoing rehabilitation. These customized experiences not only enhance engagement and motivation but also contribute to more effective

therapeutic interventions and training outcomes. To address the specific learning needs of children with disabilities, Horbova et al. [484] used SVM to analyze student performance in VR and optimize teaching methods to improve educational outcomes. Tayal et al. [485] introduced an innovative machine learning framework that integrates haptic feedback to enhance sports training. This framework enhances the training experience by offering more immersive and effective simulations. They applied the You Look Only Once algorithm with ensemble learning to analyze athlete actions and Grey Wolf Optimization to provide real-time feedback, delivering athletes realistic experiences of force, impact, and movement.

10 Discussion, challenges and future work

In this section, we briefly discuss the six previously reviewed sections, emphasizing key challenges and outlining potential directions for future research. In addition, we address several related issues.

10.1 Advanced AI-generated content representation

NeRF and 3D Gaussian are two advanced AI-generated content representations. NeRF uses MLP for implicit neural representation of the scene, while 3D Gaussian uses Gaussian sets for explicit representation of the scene. The core of NeRF representation is to represent the scene as a function of 3D locations and viewing directions, i.e., radiance field. The core of 3D Gaussian representation is to represent the scene as 3D Gaussian ellipsoids. Currently, the 3D Gaussian representation seems to be more suitable for VR applications than the NeRF representation. Because 3D Gaussian has higher rendering performance, it is suitable for real-time applications. Moreover, 3D Gaussian explicitly represents the scene as a 3D Gaussian ellipsoid, which makes 3D Gaussian more suitable for editing and interactivity. There is still some NeRF-related 3D work going on, such as 3D content editing, generation and reconstruction. However, NeRF-related methods are more often used as image/video processing tasks such as semantic segmentation, pose estimation, and compression. Although NeRF and 3D Gaussian have revolutionized scene reconstruction and novel view synthesis, some challenges still need to be addressed.

• Memory requirements. Both NeRF and 3D Gaussian memory have high memory consumption during training and rendering, especially 3D Gaussian, which can take up tens of GB. This poses a challenge for reconstructing large-scale scenes and applications in virtual reality.

• Editability of content. As an implicit representation, NeRF cannot be directly edited, hindering fast user interaction and limiting its use in interactive virtual reality. 3D Gaussian can support geometry editing to a certain extent, but support for materials, textures, and other attributes remains to be investigated.

• Few-shot problem. Most existing NeRF and 3D Gaussian-based methods typically require a large number of images from different viewpoints to model a scene accurately, which limits the use of these representations in VR.

In the future, it is critical to explore more rational and flexible 3D representations that can further improve rendering quality, achieve high rendering performance, reduce storage space, improve edibility, and reduce the amount of input data required.

10.2 Content rendering

Differentiable rendering techniques transform the traditional rasterization and ray tracing pipelines to support the backpropagation of gradient information. NeRF uses a differentiable volume rendering pipeline with typically implicit neural representations of scenes to synthesize images in new views. 3DGS uses a differentiable rasterization rendering pipeline with typically explicit 3D Gaussian representations of scenes to synthesize images in new views. There are several rendering techniques with neural models. As the pre-processing of scene contents, some materials can be modeled by neural representations, and some further operations can be carried out upon them. During the rendering procedure, neural networks can be used in path sampling for complex light transports or surface models, such as sub-surface scattering and volume rendering. For post-processing, neural networks can enhance the image quality after rendering by denoising or super-resolution technologies. The main challenges of content rendering could lie in the following directions.

• Dynamic scene rendering. These existing content rendering methods can handle dynamic scenes, but they require dynamic information during training, and for most VR applications, dynamic information cannot be obtained in advance. For most VR applications, dynamic information cannot be obtained in advance. It is a great challenge to render changing scenes efficiently and realistically based on dynamic information acquired in real time.

• High-frequency responses. Neural networks naturally tend to learn smooth and low-pass-filtered signals. However, the typical case in light transport can be of fairly high frequency.

• Denoising with complex geometry. It is non-trivial to denoise complex geometries like hairs since some auxiliary information, such as normal and depth, can be very difficult to obtain. The lighting condition is always extremely complex due to the multiple bounces of rays, and the severe aliasing issue also exists.

In the future, it is worth investigating to propose new rendering pipelines for VR to render content more efficiently and with higher quality, especially for dynamic scenes where dynamic information is not available in advance. It is also important and interesting to explore generalized neural material models in the future, and some potential advanced techniques may also have the opportunity to deal with high-frequency signals. In addition, denoising and super-resolution techniques for complex structures are of great interest.

10.3 Content generation

Current paradigms for 3D content generation based on GANs and diffusion models include lifting 2D generative models, using multi-view images as priors, and training 3D native generation models. Methods that lift 2D to 3D capitalize on advancements in the 2D generation field, thus avoiding high training costs [195,196,205,206]. However, these methods do not utilize viewpoint information, leading to poor consistency. Methods that impose multi-view images as priors improve generation quality and speed by incorporating additional viewpoint information [207, 208, 215, 221]. Training directly on 3D priors provides the best results in terms of quality and speed, though these methods are dependent on the chosen geometric representations and datasets [229, 235, 240]. Currently, methods that train directly on 3D priors show the most promising prospects due to their compatibility with existing pipelines and their geometry details. For instance, with 3D face data sampled from learned 3D GANs, deep neural networks can be trained to construct realistic 3D face models from captured single-view facial images in real time. Although these methods have approached human-level performance in single object generation tasks, there are still some serious challenges to be addressed.

• Speed of the 3D content generation. Due to the iterative denoising process, the speed of diffusion models is slow even when trained in the latent space. It is valuable to accelerate the denoising process through a good initialization or a distillation strategy.

• Efficient management of generated 3D content. To meet user needs for combination and retrieval, especially regarding the ability to understand multimodal inputs and perform cross-retrieval, efficient 3D-generated content management, especially for large-scale scenarios, remains a pressing issue.

• AI systems integrating. Combining the generated 3D content with embodied AI systems to ensure effective interaction with the physical world is also a challenging issue.

Addressing these challenges will require exploring more efficient data management techniques and more accurate 3D content modeling approaches to enhance the intelligence and utility of 3D content generation.

10.4 Physical simulation

AI-enhanced physics simulations have proven to be effective for a wide variety of physical phenomena, including fluids, soft bodies, rigid bodies, and their various combinations. These methods rely on several fundamental techniques in AI, including differentiable simulation, neural physics representation, and neural dynamics solvers. Differentiable simulation ensures that the simulation remains a continuous process, allowing seamless integration into neural networks as an efficient component with high transparency and interpretability. Many commonly used algorithms, such as SPH, MPM, and projective dynamics, have been differentiated and successfully applied in various applications, including robot training, target optimization, and more. The fidelity of the physics simulation relies on accurately tracking various Eulerian fields and Lagrangian attributes to represent the dynamic state of the system. Recent advances in implicit neural representation, such as InstantNGP [34], neural flow map [287], and neural stress field [277], offer efficient alternatives for encoding this physics information and provide the interface to neural models. This is especially valuable for complex simulations involving cross-scale scenes and multiple physics materials. The dynamics solver is the key component for encoding the laws of physics and is often the most time-consuming part of the algorithm. Current neural dynamics solvers typically transfer the current parameter space into latent space, learn to solve physics constraints using neural networks, and predict future evolution. A prevalent concern regarding the use of AI in physical simulations pertains to the explainability of AI models. Traditional physics processes are inherently transparent and grounded in well-established laws and numerical algorithms that ensure robustness, whereas AI models often operate as black boxes despite their predictive capabilities. We analyze this concern via three aforementioned techniques. First, differentiable simulations address this interpretability gap by coupling physics-constrained frameworks with analytically derived gradients, enabling end-to-end traceability of causal relationships. Second, advances in implicit neural representations demonstrate significant potential for encoding high-dimensional physical fields. Their interpretability lies in the inherent spatial structure and the compact

network for localized function approximation. Third, the neural dynamics solver remains the most opaque component, as it encodes high-dimensional approximations of complex, nonlinear physical processes. A common approach to address this is by decomposing the end-to-end learning task into modular sub-steps. By isolating intermediate states for systematic verification, this method recovers interpretable mechanistic analogs to conventional numerical solvers. To further bridge AI-enhanced physics with VR applications, several challenges await consideration.

• High-performance simulation. The interactivity of VR applications hinges on real-time simulation, which requires performance optimization through various strategies.

• Realistic reconstruction. Another crucial issue in VR is integrating virtual content with real-world scenes. Leveraging AI techniques for highly dynamic physics reconstruction and re-simulation to provide realistic and immersive physics content remains an important challenge.

• Customized physics simulation. VR users increasingly seek the ability to create and manipulate personalized physics simulations tailored to their needs, enabling more interactive experiences. Achieving this may require generative techniques to produce diverse and customizable physics phenomena.

After text, image, and video, we believe that 3D physics content will be the next key frontier driving advancements in AI development. Efficient computation through fast converging neural algorithms, neural pre-computation, and latent-space model to further improve the realism of physical phenomena reconstruction, customized physics simulation through stylization, motion control, and text-to-physics are the directions worthy of subsequent research.

10.5 Virtual characters

The development of animatable avatar and autonomous agent technologies has made significant strides, with recent advances in human motion generation, 3D avatar modeling, and autonomous character simulation. With the exploration of human structural prior and the advancements of 3D representations such as deep implicit functions [231], NeRF [30], and 3D Gaussian Splatting [35], the reconstruction of human models becomes more accurate and more affordable under sparse-view or even monocular settings. Meanwhile, the advancements in generative models have allowed for the creation of realistic and varied human motion sequences from text and audio inputs, and the implementation of complex decision-making processes in virtual environments driven by advanced large language models or large multi-modality models. Despite these advancements, several challenges remain.

• Realistic avatar. High-fidelity clothing animation and subtle facial expression generation in avatars are still challenging for existing learning-based solutions. This includes accurately modeling the complex dynamics of the human body and learning the intricate details of human motion and facial expressions, which are essential for creating believable and high-fidelity avatars.

• Emotion recognition and generation. Current technology makes it difficult to accurately capture and express complex human emotions. Although LLMs bring a wealth of prior knowledge and multimodality, they still have limitations in simulating human cognitive psychological characteristics, resulting in a lack of self-awareness in conversational scenarios.

• Autonomous behavior. Like robots and embodied intelligence, the ability of virtual autonomous agents to infer behavior depends on their deep perception and understanding of the environment, continuous knowledge learning, long and short-term memory, and accurate decision-making. How to combine real data and large models for stable and efficient modeling remains challenging. Moreover, the decision-making process of virtual agents remains opaque and difficult to control. Most autonomous agents operate as black boxes, limiting users' ability to understand or modify their behavior intuitively. This lack of explainability hinders user trust and prevents effective fine-tuning for different applications.

Looking towards the future, research and development on physically plausible avatar animation, avatars with emotion, and intelligent autonomous agents are promising. It is interesting to develop more lifelike and engaging avatars consistent with the laws of physics and human biomechanics. Future research could focus on developing algorithms that can simulate the intricate details of muscle and clothing dynamics under various interactions with surrounding humans and environments by combining state-of-the-art techniques in 3D representations such as 3DGS, differentiable simulation, and large generative models. Furthermore, future research could explore the integration of cognitive science and neuroscience into affective computing models. By leveraging cognitive architectures and brain-inspired models, avatars can develop adaptive emotional responses that vary with user interaction history, cultural background, and contextual cues. In parallel, the research of autonomous agents can further process a large number of heterogeneous data from different acquisition devices, build human psychological cognitive models to fine-tune LLMs, and improve the interpretability of the model so as to achieve dynamic real-time simulation in the VR environment. Moreover, future research could focus on designing interactive explainability frameworks where users can visually inspect an agents decision-making process, adjust behavioral parameters dynamically, and

even simulate alternative actions in real-time. VR environments can serve as a testbed for AI model behavior analysis, allowing researchers to evaluate and refine decision-making processes in immersive settings.

10.6 Interaction

Behavior recognition, interaction optimization, and perception analysis and enhancement are key research areas for the use of AI techniques in interaction. Human-centered behavior recognition involves tracking and recognizing physiological factors such as hand, gaze, and facial expressions, as well as inferring internal user intentions. Interaction optimization focuses on modeling and analyzing interaction characters of users to improve motion and manipulation in VR environments. Perception analysis and enhancement primarily address emotional computing and enhancing environmental experience.

Researchers commonly use neural networks to process complex physiological behavior recognition [424, 471] and apply regression methods such as LSTM [463], SVM [484], CNN [436] and so on to analyze user's internal psychological factors. These methods predict and recognize user characters to assist interactions. Although AI provides significant support in analyzing and enhancing user interactions, several challenges remain.

• Complex behavior recognition and prediction in high-frequency interaction scenarios. In immersive VR environments, users often perform rapid, fine-grained movements, which generate high-dimensional, time-sensitive data streams. The complexity of these interactions places significant challenges on recognition and prediction models, especially when dealing with sensor occlusion, motion blur, and signal aliasing. These issues, combined with the need for high accuracy and real-time processing, demand advanced models capable of handling the intricacies of multimodal data in such dynamic settings.

• Design of natural and immersive efficient interactions. The transition from real world to virtual world interaction methods remains challenging for multi-users, limiting the further development of VR. This calls for designing interactions that enhance naturalness and immersion, thereby increasing user acceptance and satisfaction.

Future research should focus on deeply understanding the nature of interaction to generate profound insights that enhance our comprehension of human behavior and perception. A critical area of focus will be achieving robust and seamless recognition systems tailored to personalized user experiences. This will require continuous advancements in both neural network architectures and multimodal data fusion methods, as they are essential for processing diverse and complex data streams in real time. Moreover, exploring low-latency behavior recognition, intent prediction, and interactive perception systems that maintain consistency and reliability across multiple sensory modalities (such as visual, auditory, and haptic inputs) can also make a contribution to the field. A critical area of focus will be the development of robust and seamless recognition systems tailored to personalized user experiences. Achieving this goal will require continuous advancements in both neural network architectures and multimodal data fusion techniques, as they are fundamental for processing diverse and complex data streams in real time. Additionally, exploring low-latency behavior recognition, accurate intent prediction, and interactive perception systems that ensure consistency and reliability across multiple sensory modalities will also contribute significantly to advancing the field. Exploring these areas will contribute to enhancing the intelligence of VR systems, better meeting user needs and expectations, and accelerating the adoption of VR.

10.7 Others

Data privacy and ethical issues. As VR technology advances in areas like virtual character modeling and behavior simulation, data privacy and ethical concerns are becoming more significant. To ensure compliance, three key aspects should be addressed. Ethical data collection should follow the principle of data minimization, collecting only essential data while avoiding excessive biometric data collection. Users should be clearly informed about data usage, storage, and sharing, with an option to withdraw consent at any time. Privacy protection measures should include real-time anonymization and dynamic de-identification of user behavior data. Integrating trusted execution environments in VR headsets ensures encrypted local processing of biometric data, preventing cloud transmission risks. Role-based access controls and data retention policies should be established to enhance security. Additionally, industry standards should be developed to establish a classification system for VR data and define protection levels, ensuring robust security and ethical compliance.

Computational load balance. An important issue for AI in VR is how to balance the computational load. One approach to mitigating computational load is through AI-driven optimization techniques, such as neural network compression, model pruning, and knowledge distillation. By reducing the complexity of AI models, VR applications can achieve real-time performance with lower latency and power consumption. Additionally, AI-powered predictive algorithms can anticipate user interactions, enabling pre-rendering or adaptive resource allocation to enhance responsiveness. Adaptive rendering: AI can intelligently predict user gaze direction and interaction hotspots. Techniques

such as foveated rendering optimize resource allocation by providing high-resolution images only where necessary, reducing overall computational burden. Edge computing and cloud-based AI processing can also distribute the computational burden, ensuring smooth and immersive VR experiences.

Metaverse. As an important milestone concept in the development of virtual reality, the development of the metaverse has attracted more and more attention from researchers. We believe that artificial intelligence will play a key role in shaping the future of the metaverse. First, artificial intelligence will enhance the immersive experience of the metaverse. By creating a more realistic and dynamic virtual environment, artificial intelligence can make the metaverse more attractive to users. Second, artificial intelligence algorithms can analyze user behavior and preferences to provide personalized experiences. This can lead to more efficient and effective virtual interactions, as well as better user engagement. Third, artificial intelligence can automate the process of generating and managing virtual assets such as virtual goods and services. This can reduce the cost and time required for content creation and enable more innovative and creative virtual experiences. Fourth, artificial intelligence can detect and prevent malicious activities such as hacking and fraud and protect user data and privacy. This can build trust and confidence among users and promote the healthy development of the virtual world.

11 Conclusion

With the rapid evolution of hardware and software technologies, VR technology is permeating diverse industries with unparalleled reach and depth. Users' aspirations for VR experiences have transcended basic immersion, heightened interactivity, and imagination, now embracing system intelligence, seamless interconnectivity, and continuous evolution. Recent AI breakthroughs, notably in deep learning and neural rendering, have imparted a formidable impetus to VR innovation, enriching its application scenarios and depths while offering advanced, efficient solutions tailored to the escalating diversity of user demands. This convergence heralds a new era for VR, characterized by heightened intelligence, interconnectivity, and adaptability. This review presents a comprehensive overview of AI-driven VR research. Through meticulous search and screening, we have compiled 485 pertinent papers, 93% of which were published during the AI renaissance sparked by deep neural networks, spanning 2018 to 2024. Guided by VR's core elements and pivotal research directions, we have categorized these articles, providing statistical insights and trend analyses. We delve into a technical review and discussion of these categorized works, structured around six key VR directions: advanced AI-generated content representation, content rendering, content generation, physical simulation, virtual characters, and interaction, outlining their respective research pathways. We conclude with a concise summary of these technologies, emphasizing ongoing challenges and proposing potential directions for future research. It is our hope that this review will serve as a guiding resource, shedding light on the evolving landscape of AI in virtual reality and offering valuable insights and inspiration to researchers working at the intersection of these two fields.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 62372026, 61932003, 62372025, 62272305, 62377004), Beijing Science and Technology Plan (Grant No. Z221100007722004), National Key R&D plan (Grant No. 2019YFC1521-102), and Fundamental Research Funds for the Central Universities (Grant No. 2233100028).

References

1 Zhao Q. Seizing opportunities, focusing on innovation, and developing Internet 3.0 technologies and applications. Sci Technol Herald, 2023, 41: 1
2 Wu Y, Hu K, Chen D Z, et al. AI-enhanced virtual reality in medicine: a comprehensive survey. 2024. ArXiv:2402.03093
3 Bosman K, Bosse T, Formolo D. Virtual agents for professional social skills training: an overview of the state-of-the-art. In: Proceedings of the 10th EAI International Conference on Intelligent Technologies for Interactive Entertainment, 2019. 75–84
4 Gandedkar N H, Wong M T, Darendeliler M A. Role of virtual reality (VR), augmented reality (AR) and artificial intelligence (AI) in tertiary education and research of orthodontics: an insight. Semin Orthod, 2021, 27: 69–77
5 Devagiri J S, Paheding S, Niyaz Q, et al. Augmented reality and artificial intelligence in industry: trends, tools, and future challenges. Expert Syst Appl, 2022, 207: 118002
6 Ye Z P, Xia W Y, Sun Z Y, et al. From traditional rendering to differentiable rendering: theories, methods and applications (in Chinese). Sci Sin Inform, 2021, 51: 1043–1067
7 Wu W, Wang B, Yan L Q. A survey on rendering homogeneous participating media. Comp Visual Med, 2022, 8: 177–198
8 Tewari A, Fried O, Thies J, et al. State of the art on neural rendering. Comput Graph Forum, 2020, 39: 701–727
9 Wang M, Lyu X Q, Li Y J, et al. VR content creation and exploration with deep learning: a survey. Comp Visual Med, 2020, 6: 3–28
10 Xu H, Xu J, Xu W. Survey of 3D modeling using depth cameras. Virtual Reality Intell Hardware, 2019, 1: 483–499
11 Li X, Zhang Q, Kang D, et al. Advances in 3D generation: a survey. 2024. ArXiv:2401.17807
12 Huang J, Chen J, Xu W, et al. A survey on fast simulation of elastic objects. Front Comput Sci, 2019, 13: 443–459

13 Chen Q, Wang Y, Wang H, et al. Data-driven simulation in fluids animation: a survey. Virtual Reality Intell Hardware, 2021, 3: 87–104
14 Tian Y, Zhang H, Liu Y, et al. Recovering 3D human mesh from monocular images: a survey. IEEE Trans Pattern Anal Mach Intell, 2023, 45: 15406–15425
15 Zhu W, Ma X, Ro D, et al. Human motion generation: a survey. IEEE Trans Pattern Anal Mach Intell, 2024, 46: 2430–2449
16 Sha T, Zhang W, Shen T, et al. Deep person generation: a survey from the perspective of face, pose, and cloth synthesis. ACM Comput Surv, 2023, 55: 1–37
17 Sun M, Yang D, Kou D, et al. Human 3D avatar modeling with implicit neural representation: a brief survey. In: Proceedings of the 14th International Conference on Signal Processing Systems (ICSPS), 2022. 818–827
18 Zhong J, Li D, Huang Z, et al. Data-driven crowd modeling techniques: a survey. ACM Trans Model Comput Simul, 2022, 32: 1–33
19 Wang L, Ma C, Feng X, et al. A survey on large language model based autonomous agents. Front Comput Sci, 2024, 18: 186345
20 Rahimzadeh G, Nahavandi D, Mohamed S, et al. Artificial intelligence-based motion sickness detection: a survey. In: Proceedings of the 30th IEEE International Conference on Electronics, Circuits and Systems (ICECS), 2023. 1–8
21 Li Y J, Steinicke F, Wang M. A comprehensive review of redirected walking techniques: taxonomy, methods, and future directions. J Comput Sci Technol, 2022, 37: 561–583
22 Yildirim C. A review of deep learning approaches to EEG-based classification of cybersickness in virtual reality. In: Proceedings of IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR), 2020. 351–357
23 Lv Z, Poiesi F, Dong Q, et al. Deep learning for intelligent human-computer interaction. Appl Sci, 2022, 12: 11457
24 Hirzle T, Müller F, Draxler F, et al. When XR and AI meet—a scoping review on extended reality and artificial intelligence. In: Proceedings of the CHI Conference on Human Factors in Computing Systems, 2023. 1–45
25 Oliveira T R D, Rodrigues B B, Silva M M D, et al. Virtual reality solutions employing artificial intelligence methods: a systematic literature review. ACM Comput Surv, 2023, 55: 1–29
26 Zhang R, Tsai P-S, Cryer J E, et al. Shape-from-shading: a survey. IEEE Trans Pattern Anal Machine Intell, 1999, 21: 690–706
27 Dorri A, Kanhere S S, Jurdak R. Multi-agent systems: a survey. IEEE Access, 2018, 6: 28573–28593
28 Yang S, Li T, Gong X, et al. A review on crowd simulation and modeling. Graphical Model, 2020, 111: 101081
29 Zall R, Kangavari M R. Comparative analytical survey on cognitive agents with emotional intelligence. Cogn Comput, 2022, 14: 1223–1246
30 Mildenhall B, Srinivasan P P, Tancik M, et al. NeRF: representing scenes as neural radiance fields for view synthesis. Commun ACM, 2021, 65: 99–106
31 Xu Q, Xu Z, Philip J, et al. Point-NeRF: point-based neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 5438–5448
32 Fridovich-Keil S, Yu A, Tancik M, et al. Plenoxels: radiance fields without neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 5501–5510
33 Chen A, Xu Z, Geiger A, et al. TensoRF: tensorial radiance fields. In: Proceedings of European Conference on Computer Vision, 2022. 333–350
34 Müller T, Evans A, Schied C, et al. Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans Graph, 2022, 41: 1–15
35 Kerbl B, Kopanas G, Leimkuehler T, et al. 3D Gaussian Splatting for real-time radiance field rendering. ACM Trans Graph, 2023, 42: 1–14
36 Lu T, Yu M, Xu L, et al. Scaffold-GS: structured 3D Gaussians for view-adaptive rendering. In: Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR), 2024
37 Katsumata K, Vo D M, Nakayama H. An efficient 3D Gaussian representation for monocular/multi-view dynamic scenes. 2023. ArXiv:2311.12897
38 Fan Z, Wang K, Wen K, et al. LightGaussian: unbounded 3D Gaussian compression with 15x reduction and 200+ fps. 2023. ArXiv:2311.17245
39 Kerbl B, Meuleman A, Kopanas G, et al. A hierarchical 3D Gaussian representation for real-time rendering of very large datasets. ACM Trans Graph, 2024, 43: 4
40 Li T M, Aittala M, Durand F, et al. Differentiable Monte Carlo ray tracing through edge sampling. ACM Trans Graph, 2018, 37: 1–11
41 Loubet G, Holzschuch N, Jakob W. Reparameterizing discontinuous integrands for differentiable rendering. ACM Trans Graph, 2019, 38: 1–14
42 Nimier-David M, Vicini D, Zeltner T, et al. Mitsuba 2: a retargetable forward and inverse renderer. ACM Trans Graph, 2019, 38: 1–17
43 Xu P, Bangaru S, Li T M, et al. Warped-area reparameterization of differential path integrals. ACM Trans Graph, 2023, 42: 1–18
44 Zhang C, Miller B, Yan K, et al. Path-space differentiable rendering. ACM Trans Graph, 2020, 39: 1–19
45 Zhang C, Yu Z, Zhao S. Path-space differentiable rendering of participating media. ACM Trans Graph, 2021, 40: 1–15
46 Zhou S, Chang Y, Mukai N, et al. Path-space differentiable rendering of implicit surfaces. In: Proceedings of ACM SIGGRAPH 2024 Conference, 2024
47 Zhang C, Dong Z, Doggett M, et al. Antithetic sampling for Monte Carlo differentiable rendering. ACM Trans Graph, 2021, 40: 1–12
48 Wang Y C, Wyman C, Wu L, et al. Amortizing samples in physics-based inverse rendering using ReSTIR. ACM Trans Graph, 2023, 42: 1–17
49 Chang W, Sivaram V, Nowrouzezahrai D, et al. Parameter-space restir for differentiable and inverse rendering. In: Proceedings of ACM SIGGRAPH Conference, New York, 2023
50 Fischer M, Ritschel T. Plateau-reduced differentiable path tracing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 4285–4294
51 Xing J, Luan F, Yan L Q, et al. Differentiable rendering using RGBXY derivatives and optimal transport. ACM Trans Graph, 2022, 41: 1–13
52 Xing J, Hu X, Luan F, et al. Extended path space manifolds for physically based differentiable rendering. In: Proceedings of SIGGRAPH Asia 2023 Conference, 2023. 1–11
53 Vicini D, Speierer S, Jakob W. Path replay backpropagation: differentiating light paths using constant memory and linear time. ACM Trans Graph, 2021, 40: 1–14
54 Loper M M, Black M J. Opendr: an approximate differentiable renderer. In: Proceedings of the 13th European Conference on Computer Vision, Zurich, Switzerland, 2014. 154–169
55 Kato H, Ushiku Y, Harada T. Neural 3D mesh renderer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 3907–3916
56 Liu S, Li T, Chen W, et al. Soft rasterizer: a differentiable renderer for image-based 3D reasoning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019. 7708–7717
57 Genova K, Cole F, Maschinot A, et al. Unsupervised training for 3D morphable model regression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 8377–8386
58 Lassner C, Zollhofer M. Pulsar: efficient sphere-based neural rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 1440–1449
59 Rückert D, Franke L, Stamminger M. ADOP: approximate differentiable one-pixel point rendering. ACM Trans Graph, 2022, 41: 1–14
60 Chen W, Ling H, Gao J, et al. Learning to predict 3D objects with an interpolation-based differentiable renderer. In: Proceedings of Advances in Neural Information Processing Systems, 2019. 32
61 Laine S, Hellsten J, Karras T, et al. Modular primitives for high-performance differentiable rendering. ACM Trans Graph, 2020, 39: 1–14

62 Wu L, Lee J Y, Bhattad A, et al. DIVeR: real-time and accurate neural radiance fields with deterministic integration for volume rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 16200–16209
63 Hedman P, Srinivasan P P, Mildenhall B, et al. Baking neural radiance fields for real-time view synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 5875–5884
64 Yu A, Li R, Tancik M, et al. PlenOctrees for real-time rendering of neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 5752–5761
65 Garbin S J, Kowalski M, Johnson M, et al. FastNeRF: high-fidelity neural rendering at 200fps. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 14346–14355
66 Potter M C, Wyble B, Hagmann C E, et al. Detecting meaning in RSVP at 13 ms per picture. Atten Percept Psychophys, 2014, 76: 270–279
67 Curcio C A, Allen K A. Topography of ganglion cells in human retina. J Comp Neurol, 1990, 300: 5–25
68 Guenter B, Finch M, Drucker S, et al. Foveated 3D graphics. ACM Trans Graph, 2012, 31: 1–10
69 Deng N, He Z, Ye J, et al. FoV-NeRF: foveated neural radiance fields for virtual reality. IEEE Trans Visual Comput Graph, 2022, 28: 3854–3864
70 Shi X, Wang L, Liu X, et al. Scene-aware foveated neural radiance fields. IEEE Trans Visual Comput Graph, 2025, 31: 5039–5054
71 Wang Z, Wu J, Fan R, et al. VPRF: visual perceptual radiance fields for foveated image synthesis. IEEE Trans Visual Comput Graph, 2024, 30: 7183–7192
72 Pumarola A, Corona E, Pons-Moll G, et al. D-NeRF: neural radiance fields for dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 10318–10327
73 Li Z, Niklaus S, Snavely N, et al. Neural scene flow fields for space-time view synthesis of dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 6498–6508
74 Park K, Sinha U, Barron J T, et al. NeRFies: deformable neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 5865–5874
75 Park K, Sinha U, Hedman P, et al. HyperNeRF: a higher-dimensional representation for topologically varying neural radiance fields. ACM Trans Graph, 2021, 40: 1–12
76 Xian W, Huang J B, Kopf J, et al. Space-time neural irradiance fields for free-viewpoint video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 9421–9431
77 Zhang Q, Baek S H, Rusinkiewicz S, et al. Differentiable point-based radiance fields for efficient view synthesis. In: Proceedings of SIGGRAPH Asia 2022 Conference, 2022. 1–12
78 Wang L, Zhang J, Liu X, et al. Fourier PlenOctrees for dynamic radiance field rendering in real-time. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 13524–13534
79 Song L, Chen A, Li Z, et al. NeRFPlayer: a streamable dynamic scene representation with decomposed neural radiance fields. IEEE Trans Visual Comput Graph, 2023, 29: 2732–2742
80 Shao R, Zheng Z, Tu H, et al. Tensor4D: efficient neural 4D decomposition for high-fidelity dynamic reconstruction and rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 16632–16642
81 Cao A, Johnson J. HexPlane: a fast representation for dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 130–141
82 Fridovich-Keil S, Meanti G, Warburg F R, et al. K-Planes: explicit radiance fields in space, time, and appearance. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 12479–12488
83 Liu S, Zhang X, Zhang Z, et al. Editing conditional radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 5773–5783
84 Jang W, Agapito L. CodeNeRF: disentangled neural radiance fields for object categories. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 12949–12958
85 Niemeyer M, Geiger A. Giraffe: representing scenes as compositional generative neural feature fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 11453–11464
86 Yang B, Zhang Y, Xu Y, et al. Learning object-compositional neural radiance field for editable scene rendering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 13779–13788
87 Kobayashi S, Matsumoto E, Sitzmann V. Decomposing NeRF for editing via feature field distillation. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 35: 23311–23330
88 Wang C, Chai M, He M, et al. CLIP-NeRF: text-and-image driven manipulation of neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 3835–3844
89 Chen Z, Liu Z. Relighting4d: neural relightable human from videos. In: Proceedings of European Conference on Computer Vision, 2022. 606–623
90 Rudnev V, Elgharib M, Smith W, et al. NeRF for outdoor scene relighting. In: Proceedings of European Conference on Computer Vision (ECCV), 2022
91 Barron J T, Mildenhall B, Tancik M, et al. Mip-NeRF: a multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 5855–5864
92 Barron J T, Mildenhall B, Verbin D, et al. Mip-NeRF 360: unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 5470–5479
93 Yu A, Ye V, Tancik M, et al. PixelNeRF: neural radiance fields from one or few images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 4578–4587
94 Wang Q, Wang Z, Genova K, et al. IBRNet: learning multi-view image-based rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 4690–4699
95 Chen A, Xu Z, Zhao F, et al. MVSNeRF: fast generalizable radiance field reconstruction from multi-view stereo. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 14124–14133
96 Zwicker M, Pfister H, van Baar J, et al. Ewa volume splatting. In: Proceedings of Proceedings Visualization, 2001. 29–538
97 Wu G, Yi T, Fang J, et al. 4D Gaussian Splatting for real-time dynamic scene rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 20310–20320
98 Duisterhof B P, Mandi Z, Yao Y, et al. MD-Splatting: learning metric deformation from 4D Gaussians in highly deformable scenes. 2023. ArXiv:2312.00583
99 Yang Z, Yang H, Pan Z, et al. Real-time photorealistic dynamic scene representation and rendering with 4D Gaussian Splatting. In: Proceedings of the 12th International Conference on Learning Representations, 2024
100 Kratimenos A, Lei J, Daniilidis K. DynMF: neural motion factorization for real-time dynamic view synthesis with 3D Gaussian Splatting. 2023. ArXiv:2312.00112
101 Xie T, Zong Z, Qiu Y, et al. PhysGaussian: physics-integrated 3D Gaussians for generative dynamics. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 4389–4398
102 Zhou S, Chang H, Jiang S, et al. Feature 3DGS: supercharging 3D Gaussian Splatting to enable distilled feature fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 21676–21685
103 Ye M, Danelljan M, Yu F, et al. Gaussian Grouping: segment and edit anything in 3D scenes. In: Proceedings of ECCV, 2024
104 Cen J, Fang J, Yang C, et al. Segment any 3D Gaussians. 2023. ArXiv:2312.00860
105 Liu K, Zhan F, Xu M, et al. StyleGaussian: instant 3D style transfer with Gaussian Splatting. 2024. ArXiv:2403.07807

106 Huang B, Yu Z, Chen A, et al. 2D Gaussian Splatting for geometrically accurate radiance fields. In: Proceedings of ACM SIGGRAPH 2024 Conference Papers, 2024. 1–11
107 Yu Z, Chen A, Huang B, et al. Mip-splatting: alias-free 3D Gaussian Splatting. In: Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR), 2024
108 Yan Z, Low W F, Chen Y, et al. Multi-scale 3D Gaussian Splatting for anti-aliased rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 20923–20931
109 Gao J, Gu C, Lin Y, et al. Relightable 3D Gaussian: real-time point cloud relighting with BRDF decomposition and ray tracing. 2023. ArXiv:2311.16043
110 Jiang Y, Tu J, Liu Y, et al. GaussianShader: 3D Gaussian Splatting with shading functions for reflective surfaces. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 5322–5332
111 Ma L, Agrawal V, Turki H, et al. SpecNeRF: Gaussian directional encoding for specular reflections. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 21188–21198
112 Hu B, Guo J, Chen Y, et al. DeepBRDF: a deep representation for manipulating measured BRDF. Comput Graph Forum, 2020, 39: 157–166
113 Zheng C, Zheng R, Wang R, et al. A compact representation of measured BRDFs using neural processes. ACM Trans Graph, 2022, 41: 1–15
114 Sztrajman A, Rainer G, Ritschel T, et al. Neural BRDF representation and importance sampling. Comput Graph Forum, 2021, 40: 332–346
115 Fischer M, Ritschel T. Metappearance: meta-learning for visual appearance reproduction. ACM Trans Graph, 2022, 41: 1–13
116 Gokbudak F, Sztrajman A, Zhou C, et al. Hypernetworks for generalizable BRDF estimation. 2023. ArXiv:2311.15783
117 Kuznetsov A, Hašan M, Xu Z, et al. Learning generative models for rendering specular microgeometry. ACM Trans Graph, 2019, 38: 1–14
118 Fan J, Wang B, Hasan M, et al. Neural layered BRDFs. In: Proceedings of ACM SIGGRAPH 2022 Conference, 2022. 1–8
119 Guo J, Li Z, He X, et al. MetaLayer: a meta-learned BSDF model for layered materials. ACM Trans Graph, 2023, 42: 1–15
120 TG T, Frisvad J R, Ramamoorthi R, et al. Neural BSSRDF: object appearance representation including heterogeneous subsurface scattering. 2023. ArXiv:2312.15711
121 Deschaintre V, Aittala M, Durand F, et al. Single-image SVBRDF capture with a rendering-aware deep network. ACM Trans Graph, 2018, 37: 1–15
122 Guo J, Lai S, Tu Q, et al. Ultra-high resolution SVBRDF recovery from a single image. ACM Trans Graph, 2023, 42: 1–14
123 Wang L, Zhang L, Gao F, et al. DeepBasis: hand-held single-image svBRDF capture via two-level basis material model. In: Proceedings of SIGGRAPH Asia 2023 Conference, 2023. 1–11
124 Luan F, Zhao S, Bala K, et al. Unified shape and svBRDF recovery using differentiable Monte Carlo rendering. In: Proceedings of Computer Graphics Forum, 2021. 101–113
125 Boss M, Jampani V, Kim K, et al. Two-shot spatially-varying BRDF and shape estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 3982–3991
126 Li Z, Xu Z, Ramamoorthi R, et al. Learning to reconstruct shape and spatially-varying reflectance from a single image. ACM Trans Graph, 2018, 37: 1–11
127 Rainer G, Jakob W, Ghosh A, et al. Neural BTF compression and interpolation. In: Proceedings of Computer Graphics Forum, 2019. 38
128 Rainer G, Ghosh A, Jakob W, et al. Unified neural encoding of BTFs. In: Proceedings of Computer Graphics Forum, 2020. 39
129 Kuznetsov A, Mullia K, Xu Z, et al. NeuMIP: multi-resolution neural materials. ACM Trans Graph, 2021, 40: 1–13
130 Kuznetsov A, Wang X, Mullia K, et al. Rendering neural materials on curved surfaces. In: Proceedings of ACM SIGGRAPH 2022 Conference, 2022. 1–9
131 Fan J, Wang B, Hasan M, et al. Neural biplane representation for BTF rendering and acquisition. In: Proceedings of ACM SIGGRAPH 2023 Conference Proceedings, 2023. 1–11
132 Zeltner T, Rousselle F, Weidlich A, et al. Real-time neural appearance models. ACM Trans Graph, 2024, 43: 1–17
133 Kajiya J T. The rendering equation. In: Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques, 1986. 143–150
134 Chandrasekhar S. Radiative Transfer. New York: Dover Publications, 1960
135 Müller T, Mcwilliams B, Rousselle F, et al. Neural importance sampling. ACM Trans Graph, 2019, 38: 1–19
136 Zheng Q, Zwicker M. Learning to importance sample in primary sample space. Comput Graph Forum, 2019, 38: 169–179
137 Dong H, Wang G, Li S. Neural parametric mixtures for path guiding. In: Proceedings of ACM SIGGRAPH 2023 Conference, New York, 2023
138 Huang J, Iizuka A, Tanaka H, et al. Online neural path guiding with normalized anisotropic spherical Gaussians. ACM Trans Graph, 2024, 43: 1–18
139 Zhu J, Bai Y, Xu Z, et al. Neural complex luminaires: representation and rendering. ACM Trans Graph, 2021, 40: 1–12
140 Wang Y C, Wu Y T, Li T M, et al. Learning to cluster for rendering with many lights. ACM Trans Graph, 2021, 40: 1–10
141 Bako S, Meyer M, DeRose T, et al. Offline deep importance sampling for Monte Carlo path tracing. Comput Graph Forum, 2019, 38: 527–542
142 Huo Y, Wang R, Zheng R, et al. Adaptive incident radiance field sampling and reconstruction using deep reinforcement learning. ACM Trans Graph, 2020, 39: 1–17
143 Zhu S, Xu Z, Sun T, et al. Photon-driven neural reconstruction for path guiding. ACM Trans Graph, 2022, 41: 1–15
144 Zhu S, Xu Z, Sun T, et al. Hierarchical neural reconstruction for path guiding using hybrid path and photon samples. ACM Trans Graph, 2021, 40: 1–16
145 Ren P, Wang J, Gong M, et al. Global illumination with radiance regression functions. ACM Trans Graph, 2013, 32: 1–12
146 Gao D, Mu H, Xu K. Neural global illumination: interactive indirect illumination prediction under dynamic area lights. IEEE Trans Visual Comput Graph, 2023, 29: 5325–5341
147 Müller T, Rousselle F, Nov'ak J, et al. Real-time neural radiance caching for path tracing. ACM Trans Graph, 2021, 40: 1–16
148 Hadadan S, Chen S, Zwicker M. Neural radiosity. ACM Trans Graph, 2021, 40: 1–11
149 Müller T, Rousselle F, Keller A, et al. Neural control variates. ACM Trans Graph, 2020, 39: 1–19
150 Zhu S, Xu Z, Jensen H W, et al. Deep kernel density estimation for photon mapping. Comput Graph Forum, 2020, 39: 35–45
151 Kallweit S, Müller T, Mcwilliams B, et al. Deep scattering: rendering atmospheric clouds with radiance-predicting neural networks. ACM Trans Graph, 2017, 36: 1–11
152 Hu J, Yu C, Liu H, et al. Deep real-time volumetric rendering using multi-feature fusion. In: Proceedings of ACM SIGGRAPH 2023 Conference, New York, 2023
153 Leonard L, Höhlein K, Westermann R. Learning multiple-scattering solutions for sphere-tracing of volumetric subsurface effects. Comput Graph Forum, 2021, 40: 165–178
154 Ge L, Wang B, Wang L, et al. Interactive simulation of scattering effects in participating media using a neural network model. IEEE Trans Visual Comput Graph, 2021, 27: 3123–3134
155 Vicini D, Koltun V, Jakob W. A learned shape-adaptive subsurface scattering model. ACM Trans Graph, 2019, 38: 1–15
156 KT A, Jarabo A, Aliaga C, et al. Accelerating hair rendering by learning high-order scattered radiance. Comput Graph Forum, 2023, 42: e14895

157 Kalantari N K, Bako S, Sen P. A machine learning approach for filtering Monte Carlo noise. ACM Trans Graph, 2015, 34: 1–12
158 Bako S, Vogels T, Mcwilliams B, et al. Kernel-predicting convolutional networks for denoising Monte Carlo renderings. ACM Trans Graph, 2017, 36: 1–14
159 Vogels T, Rousselle F, Mcwilliams B, et al. Denoising with kernel prediction and asymmetric loss functions. ACM Trans Graph, 2018, 37: 1–15
160 Chaitanya C R A, Kaplanyan A S, Schied C, et al. Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder. ACM Trans Graph, 2017, 36: 1–12
161 Xu B, Zhang J, Wang R, et al. Adversarial Monte Carlo denoising with conditioned auxiliary feature modulation. ACM Trans Graph, 2019, 38: 1–12
162 Yu J, Nie Y, Long C, et al. Monte Carlo denoising via auxiliary feature guided self-attention. ACM Trans Graph, 2021, 40: 1–13
163 Back J, Hua B S, Hachisuka T, et al. Self-supervised post-correction for Monte Carlo denoising. In: Proceedings of ACM SIGGRAPH 2022 Conference, 2022. 1–8
164 Gharbi M, Li T M, Aittala M, et al. Sample-based Monte Carlo denoising using a kernel-splatting network. ACM Trans Graph, 2019, 38: 1–12
165 Fu S, Lu Y, Zhang X H, et al. Monte Carlo denoising with a sparse auxiliary feature encoder. In: Proceedings of SIGGRAPH Asia 2021 Posters, 2021. 1–2
166 Işık M, Mullia K, Fisher M, et al. Interactive Monte Carlo denoising using affinity of neural features. ACM Trans Graph, 2021, 40: 1–13
167 Balint M, Wolski K, Myszkowski K, et al. Neural partitioning pyramids for denoising Monte Carlo renderings. In: Proceedings of ACM SIGGRAPH 2023 Conference, 2023. 1–11
168 Hofmann N, Hasselgren J, Munkberg J. Joint neural denoising of surfaces and volumes. Proc ACM Comput Graph Interact Tech, 2023, 6: 1–16
169 Xiao L, Nouri S, Chapman M, et al. Neural supersampling for real-time rendering. ACM Trans Graph, 2020, 39: 1–12
170 Guo Y, Chen G, Dong Y, et al. Classifier guided temporal supersampling for real-time rendering. Comput Graph Forum, 2022, 41: 237–246
171 Yang S, Zhao Y, Luo Y, et al. MNSS: neural supersampling framework for real-time rendering on mobile devices. IEEE Trans Visual Comput Graph, 2024, 30: 4271–4284
172 Zhong Z, Zhu J, Dai Y, et al. Fusesr: super resolution for real-time rendering through efficient multi-resolution fusion. In: Proceedings of SIGGRAPH Asia 2023 Conference, 2023. 1–10
173 NVIDIA. Deep learning super sampling (DLSS) technology. https://www.nvidia.com/en-us/geforce/technologies/dlss, 2023
174 AMD. AMD FidelityFX super resolution. https://www.amd.com/en/products/graphics/technologies/fidelityfx/super-resolution.html, 2021
175 Intel. Intel XE super sampling. https://www.intel.com/content/www/us/en/products/docs/arc-discrete-graphics/xess.html, 2022
176 Guo J, Fu X, Lin L, et al. Extranet: real-time extrapolated rendering for low-latency temporal supersampling. ACM Trans Graph, 2021, 40: 1–16
177 Briedis K M, Djelouah A, Meyer M, et al. Neural frame interpolation for rendered content. ACM Trans Graph, 2021, 40: 1–13
178 Wu Z, Zuo C, Huo Y, et al. Adaptive recurrent frame prediction with learnable motion vectors. In: Proceedings of SIGGRAPH Asia 2023 Conference, 2023. 1–11
179 Briedis K M, Djelouah A, Ortiz R, et al. Kernel-based frame interpolation for spatio-temporally adaptive rendering. In: Proceedings of ACM SIGGRAPH 2023 Conference, 2023. 1–11
180 Wu S, Kim S, Zeng Z, et al. Extrass: a framework for joint spatial super sampling and frame extrapolation. In: Proceedings of SIGGRAPH Asia 2023 Conference, 2023. 1–11
181 He R, Zhou S, Sun Y, et al. Low-latency space-time supersampling for real-time rendering. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2024. 38: 2103–2111
182 Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets. In: Proceedings of Advances in Neural Information Processing Systems, 2014. 27
183 Achlioptas P, Diamanti O, Mitliagkas I, et al. Learning representations and generative models for 3D point clouds. In: Proceedings of International Conference on Machine Learning, 2018. 40–49
184 Knyaz V A, Kniaz V V, Remondino F. Image-to-voxel model translation with conditional adversarial networks. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops, 2018
185 Luo A, Li T, Zhang W H, et al. Surfgen: adversarial 3D shape synthesis with explicit surface discriminators. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 16238–16248
186 Schwarz K, Liao Y, Niemeyer M, et al. Graf: generative radiance fields for 3D-aware image synthesis. In: Proceedings of Advances in Neural Information Processing Systems, 2020. 33: 20154–20166
187 Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models. In: Proceedings of Advances in Neural Information Processing Systems, 2020. 33: 6840–6851
188 Luo S, Hu W. Diffusion probabilistic models for 3D point cloud generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 2837–2845
189 Liu Z, Feng Y, Black M J, et al. Meshdiffusion: score-based generative 3d mesh modeling. In: Proceedings of the 11th International Conference on Learning Representations, 2023
190 Jun H, Nichol A. Shap-E: generating conditional 3D implicit functions. 2023. ArXiv:2305.02463
191 Shim J, Kang C, Joo K. Diffusion-based signed distance fields for 3D shape generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 20887–20897
192 Ramesh A, Pavlov M, Goh G, et al. Zero-shot text-to-image generation. In: Proceedings of the 38th International Conference on Machine Learning, 2021. 8821–8831
193 Saharia C, Chan W, Saxena S, et al. Photorealistic text-to-image diffusion models with deep language understanding. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 35: 36479–36494
194 Rombach R, Blattmann A, Lorenz D, et al. High-resolution image synthesis with latent diffusion models. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Los Alamitos, 2022. 10674–10685
195 Poole B, Jain A, Barron J T, et al. Dreamfusion: text-to-3D using 2D diffusion. In: Proceedings of the 11th International Conference on Learning Representations, 2023
196 Lin C H, Gao J, Tang L, et al. Magic3d: high-resolution text-to-3D content creation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 300–309
197 Yu C, Zhou Q, Li J, et al. Points-to-3d: bridging the gap between sparse points and shape-controllable text-to-3D generation. In: Proceedings of the 31st ACM International Conference on Multimedia, New York, 2023. 6841–6850
198 Zhu J, Zhuang P, Koyejo S. HIFA: high-fidelity text-to-3D generation with advanced diffusion guidance. In: Proceedings of the 12th International Conference on Learning Representations, 2024
199 Huang Y, Wang J, Shi Y, et al. Dreamtime: an improved optimization strategy for text-to-3D content creation. 2023. ArXiv:2306.12422
200 Wu J, Gao X, Liu X, et al. Hd-fusion: detailed text-to-3D generation leveraging multiple noise estimation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024. 3202–3211
201 Chen Z, Wang F, Wang Y, et al. Text-to-3D using Gaussian splatting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 21401–21412

202 Shen T, Gao J, Yin K, et al. Deep marching tetrahedra: a hybrid representation for high-resolution 3D shape synthesis. In: Proceedings of Advances in Neural Information Processing Systems, 2021. 34: 6087–6101
203 Seo J, Jang W, Kwak M S, et al. Let 2D diffusion model know 3D-consistency for robust text-to-3D generation. In: Proceedings of the 12th International Conference on Learning Representations, 2024
204 Li W, Chen R, Chen X, et al. Sweetdreamer: aligning geometric priors in 2D diffusion for consistent text-to-3D. In: Proceedings of the 12th International Conference on Learning Representations, 2024
205 Chen R, Chen Y, Jiao N, et al. Fantasia3D: disentangling geometry and appearance for high-quality text-to-3D content creation. In: Proceedings of 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Los Alamitos, 2023. 22189–22199
206 Wang Z, Lu C, Wang Y, et al. Prolificdreamer: high-fidelity and diverse text-to-3D generation with variational score distillation. In: Proceedings of Advances in Neural Information Processing Systems, 2024. 36
207 Liu R, Wu R, van Hoorick B, et al. Zero-1-to-3: zero-shot one image to 3D object. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023. 9298–9309
208 Qian G, Mai J, Hamdi A, et al. Magic123: one image to high-quality 3D object generation using both 2D and 3D diffusion priors. In: Proceedings of the 12th International Conference on Learning Representations, 2024
209 Zhang Y, Yu J, Song Z, et al. Optimized view and geometry distillation from multi-view diffuser. 2023. ArXiv:2312.06198
210 Tang J, Ren J, Zhou H, et al. Dreamgaussian: generative Gaussian Splatting for efficient 3D content creation. In: Proceedings of the 12th International Conference on Learning Representations, 2024
211 Sun J, Zhang B, Shao R, et al. Dreamcraft3D: hierarchical 3D generation with bootstrapped diffusion prior. In: Proceedings of the 12th International Conference on Learning Representations, 2024
212 Shi Y, Wang P, Ye J, et al. MVDream: multi-view diffusion for 3D generation. In: Proceedings of the 12th International Conference on Learning Representations, 2024
213 Liu Y, Lin C, Zeng Z, et al. SyncDreamer: generating multiview-consistent images from a single-view image. In: Proceedings of the 12th International Conference on Learning Representations, 2024
214 Wang P, Shi Y. Imagedream: image-prompt multi-view diffusion for 3D generation. 2023. ArXiv:2312.02201
215 Shi R, Chen H, Zhang Z, et al. Zero123++: a single image to consistent multi-view diffusion base model. 2023. ArXiv:2310.15110
216 Li S, Li C, Zhu W, et al. 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, New York, 2023
217 Long X, Guo Y C, Lin C, et al. Wonder3D: single image to 3D using cross-domain diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 9970–9980
218 Qiu L, Chen G, Gu X, et al. Richdreamer: a generalizable normal-depth diffusion model for detail richness in text-to-3D. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 9914–9925
219 Blattmann A, Dockhorn T, Kulal S, et al. Stable video diffusion: scaling latent video diffusion models to large datasets. 2023. ArXiv:2311.15127
220 Long X, Lin C, Wang P, et al. Sparseneus: fast generalizable neural surface reconstruction from sparse views. In: Proceedings of Computer Vision—ECCV 2022. Cham: Springer Nature Switzerland, 2022. 210–227
221 Liu M, Xu C, Jin H, et al. One-2-3-45: any single image to 3D mesh in 45 seconds without per-shape optimization. In: Proceedings of Advances in Neural Information Processing Systems, 2024. 36
222 Hong Y, Zhang K, Gu J, et al. LRM: large reconstruction model for single image to 3D. In: Proceedings of the 12th International Conference on Learning Representations, 2024
223 Wang P, Tan H, Bi S, et al. PF-LRM: pose-free large reconstruction model for joint pose and shape prediction. In: Proceedings of the 12th International Conference on Learning Representations, 2024
224 Xu Y, Tan H, Luan F, et al. DMV3d: denoising multi-view diffusion using 3D large reconstruction model. In: Proceedings of the 12th International Conference on Learning Representations, 2024
225 Zou Z X, Yu Z, Guo Y C, et al. 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, 2024. 10324–10335
226 Xie D, Bi S, Shu Z, et al. LRM-zero: training large reconstruction models with synthesized data. 2024. ArXiv:2406.09371
227 Nichol A, Jun H, Dhariwal P, et al. Point-e: a system for generating 3D point clouds from complex prompts. 2022. ArXiv:2212.08751
228 Nash C, Ganin Y, Eslami S A, et al. Polygen: an autoregressive generative model of 3D meshes. In: Proceedings of International Conference on Machine Learning, 2020. 7220–7229
229 Siddiqui Y, Alliegro A, Artemov A, et al. MeshGPT: generating triangle meshes with decoder-only transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 19615–19625
230 Ren X, Huang J, Zeng X, et al. XCube: large-scale 3D generative modeling using sparse voxel hierarchies. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 4209–4219
231 Park J, Florence P, Straub J, et al. DeepSDF: learning continuous signed distance functions for shape representation. In: Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Los Alamitos, 2019. 165–174
232 Yariv L, Puny O, Gafni O, et al. Mosaic-SDF for 3D generative models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 4630–4639
233 Cheng Y, Lee H, Tulyakov S, et al. SDFfusion: multimodal 3D shape completion, reconstruction, and generation. In: Proceedings of 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Los Alamitos, 2023. 4456–4465
234 Zheng X Y, Pan H, Wang P S, et al. Locally attentional SDF diffusion for controllable 3D shape generation. ACM Trans Graph, 2023, 42: 1–13
235 Yin F, Chen X, Zhang C, et al. ShapeGPT: 3D shape generation with a unified multi-modal language model. 2023. ArXiv:2311.17618
236 Chang A X, Funkhouser T, Guibas L, et al. Shapenet: an information-rich 3D model repository. 2015. ArXiv:1512.03012
237 Gupta A, Xiong W, Nie Y, et al. 3dgen: triplane latent diffusion for textured mesh generation. 2023. ArXiv:2303.05371
238 Zhang B, Tang J, Nießner M, et al. 3DShape2VecSet: a 3D shape representation for neural fields and generative diffusion models. ACM Trans Graph, 2023, 42: 1–16
239 Zhao Z, Liu W, Chen X, et al. Michelangelo: conditional 3D shape generation based on shape-image-text aligned latent representation. In: Proceedings of Advances in Neural Information Processing Systems, 2023
240 Zhang L, Wang Z, Zhang Q, et al. CLAY: a controllable large-scale generative model for creating high-quality 3D assets. ACM Trans Graph, 2024, 43: 1–20
241 Li W, Liu J, Chen R, et al. Craftsman: high-fidelity mesh generation with 3D native generation and interactive geometry refiner. 2024. ArXiv:2405.14979
242 Wu S, Lin Y, Zeng Y, et al. Direct3D: scalable image-to-3D generation via 3D latent diffusion transformer. In: Proceedings of Advances in Neural Information Processing Systems, 2024. 37: 121859–121881
243 Chen Z, Wang G, Liu Z. SceneDreamer: unbounded 3D scene generation from 2D image collections. IEEE Trans Pattern Anal Mach Intell, 2023, 45: 15562–15576
244 Bahmani S, Park J J, Paschalidou D, et al. CC3D: layout-conditioned generation of compositional 3D scenes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023. 7171–7181
245 Zhang Q, Xu Y, Shen Y, et al. BerfScene: BEV-conditioned equivariant radiance fields for infinite 3D scene generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 6839–6849
246 Xu Y, Chai M, Shi Z, et al. Discoscene: spatially disentangled generative radiance fields for controllable 3D-aware scene synthesis. In:

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 4402–4412
247 Kim S W, Brown B, Yin K, et al. Neuralfield-lDM: scene generation with hierarchical latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 8496–8506
248 Zheng Y, Li X, Nagano K, et al. A unified approach for text-and image-guided 4D scene generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 7300–7309
249 Zhang J, Li X, Wan Z, et al. Text2NeRF: text-driven 3D scene generation with neural radiance fields. IEEE Trans Visual Comput Graph, 2024, 30: 7749–7762
250 Zhang S, Zhang Y, Zheng Q, et al. 3D-scenedreamer: text-driven 3D-consistent scene generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 10170–10180
251 Po R, Wetzstein G. Compositional 3D scene generation using locally conditioned diffusion. In: Proceedings of International Conference on 3D Vision (3DV), 2024. 651–663
252 Hudson D A, Zitnick L. Compositional transformers for scene generation. In: Proceedings of Advances in Neural Information Processing Systems, 2021. 34: 9506–9520
253 Paschalidou D, Kar A, Shugrina M, et al. ATISS: autoregressive transformers for indoor scene synthesis. In: Proceedings of Advances in Neural Information Processing Systems, 2021. 34: 12013–12026
254 Yi H, Huang C H P, Tripathi S, et al. MIME: human-aware 3D scene generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 12965–12976
255 Tang J, Nie Y, Markhasin L, et al. DiffuScene: denoising diffusion models for generative indoor scene synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 20507–20518
256 Zhai G, Örnek E P, Wu S C, et al. CommonScenes: generating commonsense 3D indoor scenes with scene graphs. In: Proceedings of Advances in Neural Information Processing Systems, 2024. 36
257 Feng W, Zhu W, Fu T J, et al. LayoutGPT: compositional visual planning and generation with large language models. In: Proceedings of Advances in Neural Information Processing Systems, 2024. 36
258 Zesch R S, Modi V, Sueda S, et al. Neural collision fields for triangle primitives. In: Proceedings of SIGGRAPH Asia 2023 Conference, New York, 2023
259 Qiao Y L, Liang J, Koltun V, et al. Efficient differentiable simulation of articulated bodies. In: Proceedings of ICML, 2021
260 Qiao Y, Liang J, Koltun V, et al. Differentiable simulation of soft multi-body systems. In: Proceedings of Advances in Neural Information Processing Systems, 2021. 34: 17123–17135
261 Geilinger M, Hahn D, Zehnder J, et al. ADD: analytically differentiable dynamics for multi-body systems with frictional contact. ACM Trans Graph, 2020, 39: 1–15
262 Ehsani K, Tulsiani S, Gupta S, et al. Use the force, Luke! Learning to predict physical forces by simulating effects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 224–233
263 Groth O, Fuchs F B, Posner I, et al. Shapestacks: learning vision-based physical intuition for generalised object stacking. In: Proceedings of the European Conference on Computer Vision, 2018. 702–717
264 Lyu Q, Chai M, Chen X, et al. Real-time hair simulation with neural interpolation. IEEE Trans Visual Comput Graph, 2020, 28: 1894–1905
265 Wang Z, Nam G, Stuyck T, et al. Neuwigs: a neural dynamic model for volumetric hair capture and animation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 8641–8651
266 Liang J, Lin M, Koltun V. Differentiable cloth simulation for inverse problems. In: Proceedings of Advances in Neural Information Processing Systems, 2019
267 Li Y, Du T, Wu K, et al. DiffCloth: differentiable cloth simulation with dry frictional contact. ACM Trans Graph, 2023, 42: 1–20
268 Santesteban I, Thuerey N, Otaduy M A, et al. Self-supervised collision handling via generative 3D garment models for virtual try-on. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 11763–11773
269 Bertiche H, Madadi M, Escalera S. PBNS: physically based neural simulation for unsupervised garment pose space deformation. ACM Trans Graph, 2021, 40: 1–14
270 Bertiche H, Madadi M, Escalera S. Neural cloth simulation. ACM Trans Graph, 2022, 41: 1–14
271 Zhang M, Wang T Y, Ceylan D, et al. Dynamic neural garments. ACM Trans Graph, 2021, 40: 1–15
272 Xiang D, Bagautdinov T, Stuyck T, et al. Dressing avatars: deep photorealistic appearance for physically simulated clothing. ACM Trans Graph, 2022, 41: 1–15
273 Wang T Y, Shao T, Fu K, et al. Learning an intrinsic garment space for interactive authoring of garment animation. ACM Trans Graph, 2019, 38: 1–12
274 Fulton L, Modi V, Duvenaud D, et al. Latent-space dynamics for reduced deformable simulation. Comput Graph Forum, 2019, 38: 379–391
275 Du T, Wu K, Ma P, et al. Diffpd: differentiable projective dynamics. ACM Trans Graph, 2021, 41: 1–21
276 Huang Z, Tozoni D C, Gjoka A, et al. Differentiable solver for time-dependent deformation problems with contact. ACM Trans Graph, 2024, 43: 1–30
277 Zong Z, Li X, Li M, et al. Neural stress fields for reduced-order elastoplasticity and fracture. In: Proceedings of SIGGRAPH Asia 2023 Conference, 2023. 1–11
278 Romero C, Casas D, Chiaramonte M, et al. Learning contact deformations with general collider descriptors. In: Proceedings of SIGGRAPH Asia 2023 Conference, 2023. 1–10
279 Yang L, Kim B, Zoss G, et al. Implicit neural representation for physics-driven actuated soft bodies. ACM Trans Graph, 2022, 41: 1–10
280 Feng Y, Shang Y, Li X, et al. Pie-NeRF: physics-based interactive elastodynamics with NeRF. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 4450–4461
281 Zhang T, Yu H X, Wu R, et al. PhysDreamer: physics-based interaction with 3D objects via video generation. 2024. ArXiv:2404.13026
282 Takahashi T, Liang J, Qiao Y L, et al. Differentiable fluids with solid coupling for learning and control. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2021. 35: 6138–6146
283 Chu M, Thuerey N, Seidel H P, et al. Learning meaningful controls for fluids. ACM Trans Graph, 2021, 40: 1–13
284 Yang C, Yang X, Xiao X. Data-driven projection method in fluid simulation. Comput Animation Virtual, 2016, 27: 415–424
285 Xiao X, Wang H, Yang X. A CNN-based flow correction method for fast preview. Comput Graph Forum, 2019, 38: 431–440
286 Xiao X, Zhou Y, Wang H, et al. A novel CNN-based Poisson solver for fluid simulation. IEEE Trans Visual Comput Graph, 2020, 26: 1454–1465
287 Deng Y, Yu H X, Zhang D, et al. Fluid simulation on neural flow maps. ACM Trans Graph, 2023, 42: 1–21
288 Eckert M L, Um K, Thuerey N. ScalarFlow: a large-scale volumetric data set of real-world scalar transport flows for computer animation and machine learning. ACM Trans Graph, 2019, 38: 1–16
289 Chu M, Liu L, Zheng Q, et al. Physics informed neural fields for smoke reconstruction with sparse data. ACM Trans Graph, 2022, 41: 1–14
290 Kim B, Azevedo V C, Gross M, et al. Transport-based neural style transfer for smoke simulations. ACM Trans Graph, 2019, 38: 1–11
291 Guo J, Li M, Zong Z, et al. Volumetric appearance stylization with stylizing kernel prediction network. ACM Trans Graph, 2021, 40: 1–15
292 Aurand J, Ortiz R, Nauer S, et al. Efficient neural style transfer for volumetric simulations. ACM Trans Graph, 2022, 41: 1–10
293 Ummenhofer B, Prantl L, Thuerey N, et al. Lagrangian fluid simulation with continuous convolutions. In: Proceedings of International

Conference on Learning Representations, 2019
294 Shao Y, Loy C C, Dai B. Transformer with implicit edges for particle-based physics simulation. In: Proceedings of European Conference on Computer Vision, 2022. 549–564
295 Li J, Gao Y, Dai J, et al. MPMNet: a data-driven MPM framework for dynamic fluid-solid interaction. IEEE Trans Visual Comput Graph, 2024, 30: 4694–4708
296 Franz E, Solenthaler B, Thuerey N. Global transport for fluid reconstruction with learned self-supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 1632–1642
297 Guan S, Deng H, Wang Y, et al. NeuroFluid: fluid dynamics grounding with particle-driven neural radiance fields. In: Proceedings of International Conference on Machine Learning, 2022. 7919–7929
298 Xiao S, Tong C, Zhang Q, et al. Laplacian projection based global physical prior smoke reconstruction. IEEE Trans Visual Comput Graph, 2024, 30: 7657–7671
299 Roy B, Poulin P, Paquette E. Neural UpFlow: a scene flow learning approach to increase the apparent resolution of particlebased liquids. Proc ACM Comput Graph Interact Tech, 2021, 4: 1–26
300 Prantl L, Ummenhofer B, Koltun V, et al. Guaranteed conservation of momentum for learning particle-based fluid dynamics. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 35: 6901–6913
301 Kim B, Azevedo V C, Gross M, et al. Lagrangian neural style transfer for fluids. ACM Trans Graph, 2020, 39: 1–10
302 Ma P, Tian Y, Pan Z, et al. Fluid directed rigid body control using deep reinforcement learning. ACM Trans Graph, 2018, 37: 1–11
303 Ren B, Ye X, Pan Z, et al. Versatile control of fluid-directed solid objects using multi-task reinforcement learning. ACM Trans Graph, 2023, 42: 1–14
304 Li Z, Xu Q, Ye X, et al. DiffFR: differentiable SPH-based fluid-rigid coupling for rigid body control. ACM Trans Graph, 2023, 42: 1–17
305 Yan G, Chen Z, Yang J, et al. Interactive liquid splash modeling by user sketches. ACM Trans Graph, 2020, 39: 1–13
306 Feng Y, Feng X, Shang Y, et al. Gaussian splashing: dynamic fluid synthesis with Gaussian Splatting. 2024. ArXiv:2401.15318
307 Hu Y, Anderson L, Li T M, et al. diffTaichi: differentiable programming for physical simulation. In: Proceedings of International Conference on Learning Representations, 2020
308 Hu Y, Liu J, Spielberg A, et al. Chainqueen: a real-time differentiable physical simulator for soft robotics. In: Proceedings of International Conference on Robotics and Automation (ICRA), 2019. 6265–6271
309 Sanchez-Gonzalez A, Godwin J, Pfaff T, et al. Learning to simulate complex physics with graph networks. In: Proceedings of International Conference on Machine Learning, 2020. 8459–8468
310 Hennigh O, Narasimhan S, Nabian M A, et al. Nvidia SimNet: an AI-accelerated multi-physics simulation framework. In: Proceedings of International Conference on Computational Science, 2021. 447–461
311 Wang H, Yu T, Yang T, et al. Neural physical simulation with multi-resolution hash grid encoding. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2024, 38: 5410–5418
312 Li X, Qiao Y L, Chen P Y, et al. PAC-NeRF: physics augmented continuum neural radiance fields for geometry-agnostic system identification. In: Proceedings of the 11th International Conference on Learning Representations, 2023
313 Zhang Y, Yu W, Liu C K, et al. Learning to manipulate amorphous materials. ACM Trans Graph, 2020, 39: 1–11
314 Jiang Y, Yu C, Xie T, et al. VR-GS: a physical dynamics-aware interactive Gaussian Splatting system in virtual reality. In: Proceedings of ACM SIGGRAPH 2024 Conference, 2024
315 Huang Z, Chen F, Pu Y, et al. DiffVL: scaling up soft body manipulation using vision-language driven differentiable physics. In: Proceedings of Advances in Neural Information Processing Systems, 2023. 36: 29875–29900
316 Qiu R Z, Yang G, Zeng W, et al. Feature splatting: language-driven physics-based scene synthesis and editing. 2024. ArXiv:2404.01223
317 Guo C, Zuo X, Wang S, et al. Action2Motion: conditioned generation of 3d human motions. In: Proceedings of the 28th ACM International Conference on Multimedia, 2020. 2021–2029
318 Petrovich M, Black M J, Varol G. Action-conditioned 3D human motion synthesis with transformer VAE. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 10985–10995
319 Yu P, Zhao Y, Li C, et al. Structure-aware human-action generation. In: Proceedings of the 16th European Conference on Computer Vision–ECCV 2020, Glasgow, 2020. 18–34
320 Degardin B, Neves J, Lopes V, et al. Generative adversarial graph convolutional networks for human action synthesis. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022. 1150–1159
321 Lu Q, Zhang Y, Lu M, et al. Action-conditioned on-demand motion generation. In: Proceedings of the 30th ACM International Conference on Multimedia, 2022. 2249–2257
322 Lucas T, Baradel F, Weinzaepfel P, et al. PoseGPT: quantization-based 3D human motion generation and forecasting. In: Proceedings of European Conference on Computer Vision, 2022. 417–435
323 Lee T, Moon G, Lee K M. MultiAct: long-term 3D human motion generation from multiple action labels. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2023. 37: 1231–1239
324 Ahn H, Ha T, Choi Y, et al. Text2Action: generative adversarial synthesis from language to action. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA), 2018. 5915–5920
325 Ahuja C, Morency L P. Language2pose: natural language grounded pose forecasting. In: Proceedings of International Conference on 3D Vision (3DV), 2019. 719–728
326 Petrovich M, Black M J, Varol G. TEMOS: generating diverse human motions from textual descriptions. In: Proceedings of European Conference on Computer Vision, 2022. 480–497
327 Guo C, Zuo X, Wang S, et al. TM2T: stochastic and tokenized modeling for the reciprocal generation of 3D human motions and texts. In: Proceedings of European Conference on Computer Vision, 2022. 580–597
328 Hong F, Zhang M, Pan L, et al. AvatarCLIP: zero-shot text-driven generation and animation of 3D avatars. ACM Trans Graph, 2022, 41: 1–19
329 Tevet G, Gordon B, Hertz A, et al. MotionCLIP: exposing human motion generation to clip space. In: Proceedings of European Conference on Computer Vision, 2022. 358–374
330 Dabral R, Mughal M H, Golyanik V, et al. MoFusion: a framework for denoising-diffusion-based motion synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 9760–9770
331 Ginosar S, Bar A, Kohavi G, et al. Learning individual styles of conversational gesture. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 3497–3506
332 Li J, Kang D, Pei W, et al. Audio2Gestures: generating diverse gestures from speech audio with conditional variational autoencoders. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 11293–11302
333 Yoon Y, Cha B, Lee J H, et al. Speech gesture generation from the trimodal context of text, audio, and speaker identity. ACM Trans Graph, 2020, 39: 1–16
334 Chen K, Tan Z, Lei J, et al. ChoreoMaster: choreography-oriented music-driven dance synthesis. ACM Trans Graph, 2021, 40: 1–13
335 Gao J, Pu J, Zhang H, et al. Pc-dance: posture-controllable music-driven dance synthesis. In: Proceedings of the 30th ACM International Conference on Multimedia, 2022. 1261–1269
336 Kim J, Oh H, Kim S, et al. A brand new dance partner: music-conditioned pluralistic dancing controlled by multiple dance genres. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 3490–3500
337 Tseng J, Castellon R, Liu K. EDGE: editable dance generation from music. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 448–458

338 Liu J, Shahroudy A, Perez M, et al. NTU RGB+D 120: a large-scale benchmark for 3D human activity understanding. IEEE Trans Pattern Anal Mach Intell, 2019, 42: 2684–2701
339 Ji Y, Xu F, Yang Y, et al. A large-scale RGB-D database for arbitrary-view human action recognition. In: Proceedings of the 26th ACM International Conference on Multimedia, 2018. 1510–1518
340 Ionescu C, Papava D, Olaru V, et al. Human3.6M: large scale datasets and predictive methods for 3D human sensing in natural environments. IEEE Trans Pattern Anal Mach Intell, 2013, 36: 1325–1339
341 Shahroudy A, Liu J, Ng T T, et al. NTU RGB+ D: a large scale dataset for 3D human activity analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 1010–1019
342 Punnakkal A R, Chandrasekaran A, Athanasiou N, et al. Babel: bodies, action and behavior with English labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 722–731
343 Taheri O, Ghorbani N, Black M J, et al. Grab: a dataset of whole-body human grasping of objects. In: Proceedings of the 16th European Conference on Computer Vision–ECCV 2020, Glasgow, 2020. 581–600
344 Xu J, Mei T, Yao T, et al. MSR-VTT: a large video description dataset for bridging video and language. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 5288–5296
345 Plappert M, Mandery C, Asfour T. The KIT Motion-language dataset. Big Data, 2016, 4: 236–252
346 Guo C, Zou S, Zuo X, et al. Generating diverse and natural 3D human motions from text. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 5152–5161
347 Mahmood N, Ghorbani N, Troje N F, et al. Amass: archive of motion capture as surface shapes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019. 5442–5451
348 Li R, Yang S, Ross D A, et al. AI choreographer: music conditioned 3D dance generation with AIST++. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 13401–13412
349 Ferstl Y, McDonnell R. Investigating the use of recurrent motion modelling for speech gesture generation. In: Proceedings of the 18th International Conference on Intelligent Virtual Agents, 2018. 93–98
350 Schonberger J L, Frahm J M. Structure-from-motion revisited. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 4104–4113
351 Collet A, Chuang M, Sweeney P, et al. High-quality streamable free-viewpoint video. ACM Trans Graph, 2015, 34: 1–13
352 Guo K, Lincoln P, Davidson P, et al. The relightables: volumetric performance capture of humans with realistic relighting. ACM Trans Graph, 2019, 38: 1–19
353 Jiang Y, Jiang S, Sun G, et al. Neuralhofusion: neural volumetric rendering under human-object interactions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 6155–6165
354 Lin W, Zheng C, Yong J H, et al. Occlusionfusion: occlusion-aware motion estimation for real-time dynamic 3d reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 1736–1745
355 Xu L, Su Z, Han L, et al. UnstructuredFusion: realtime 4D geometry and texture reconstruction using commercial RGBD cameras. IEEE Trans Pattern Anal Mach Intell, 2019, 42: 2508–2522
356 Zheng Y, Shao R, Zhang Y, et al. Deepmulticap: performance capture of multiple characters using sparse multiview cameras. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 6239–6249
357 Schönberger J L, Zheng E, Frahm J M, et al. Pixelwise view selection for unstructured multi-view stereo. In: Proceedings of the 14th European Conference on Computer Vision–ECCV 2016, Amsterdam, 2016. 501–518
358 Zheng E, Dunn E, Jojic V, et al. PatchMatch based joint view selection and depthmap estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014. 1510–1517
359 Newcombe R A, Fox D, Seitz S M. DynamicFusion: reconstruction and tracking of non-rigid scenes in real-time. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015. 343–352
360 Yu T, Zheng Z, Guo K, et al. DoubleFusion: real-time capture of human performances with inner body shapes from a single depth sensor. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 7287–7296
361 Loper M, Mahmood N, Romero J, et al. SMPL: a skinned multi-person linear model. In: Proceedings of Seminal Graphics Papers: Pushing the Boundaries, 2023. 851–866
362 Dou M, Davidson P, Fanello S R, et al. Motion2Fusion: real-time volumetric performance capture. ACM Trans Graph, 2017, 36: 1–16
363 Su Z, Xu L, Zheng Z, et al. RobustFusion: human volumetric capture with data-driven visual cues using a RGBD camera. In: Proceedings of 16th European Conference on Computer Vision–ECCV 2020, Glasgow, 2020. 246–264
364 Saito S, Huang Z, Natsume R, et al. PIFu: pixel-aligned implicit function for high-resolution clothed human digitization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019. 2304–2314
365 Shao R, Zhang H, Zhang H, et al. DoubleField: bridging the neural surface and radiance fields for high-fidelity human reconstruction and rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 15872–15882
366 Shao R, Zheng Z, Zhang H, et al. DiffuStereo: high quality human reconstruction via diffusion-based stereo using sparse cameras. In: Proceedings of European Conference on Computer Vision, 2022. 702–720
367 Pavlakos G, Choutas V, Ghorbani N, et al. Expressive body capture: 3D hands, face, and body from a single image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 10975–10985
368 Peng S, Zhang Y, Xu Y, et al. Neural Body: implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 9054–9063
369 Wang S, Mihajlovic M, Ma Q, et al. MetaAvatar: learning animatable clothed human models from few depth images. In: Proceedings of Advances in Neural Information Processing Systems, 2021. 34: 2810–2822
370 Tiwari G, Sarafianos N, Tung T, et al. Neural-GIF: neural generalized implicit functions for animating people in clothing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 11708–11718
371 Mihajlovic M, Zhang Y, Black M J, et al. LEAP: learning articulated occupancy of people. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 10461–10471
372 Saito S, Yang J, Ma Q, et al. SCANimate: weakly supervised learning of skinned clothed avatar networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 2886–2897
373 Chen X, Zheng Y, Black M J, et al. SNARF: differentiable forward skinning for animating non-rigid neural implicit shapes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 11594–11604
374 Bagautdinov T, Wu C, Simon T, et al. Driving-signal aware full-body avatars. ACM Trans Graph, 2021, 40: 1–17
375 Peng S, Dong J, Wang Q, et al. Animatable neural radiance fields for modeling dynamic human bodies. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 14314–14323
376 Zheng Z, Zhao X, Zhang H, et al. AvatarRex: real-time expressive full-body avatars. ACM Trans Graph, 2023, 42: 1–19
377 Weng C Y, Curless B, Srinivasan P P, et al. HumanNeRF: free-viewpoint rendering of moving people from monocular video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 16210–16220
378 Liu L, Habermann M, Rudnev V, et al. Neural Actor: neural free-view synthesis of human actors with pose control. ACM Trans Graph, 2021, 40: 1–16
379 Li Z, Zheng Z, Wang L, et al. Animatable Gaussians: learning pose-dependent Gaussian maps for high-fidelity human avatar modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 19711–19722
380 Jackson A S, Bulat A, Argyriou V, et al. Large pose 3D face reconstruction from a single image via direct volumetric CNN regression. In: Proceedings of the IEEE International Conference on Computer Vision, 2017. 1031–1039

381 Feng Y, Wu F, Shao X, et al. Joint 3D face reconstruction and dense alignment with position map regression network. In: Proceedings of the European Conference on Computer Vision (ECCV), 2018. 534–551
382 Jiang L, Zhang J, Deng B, et al. 3D face reconstruction with geometry details from a single image. IEEE Trans Image Process, 2018, 27: 4756–4770
383 Wu F, Bao L, Chen Y, et al. MVF-net: multi-view 3D face morphable model regression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 959–968
384 Bai Z, Cui Z, Rahim J A, et al. Deep facial non-rigid multi-view stereo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 5850–5860
385 Yenamandra T, Tewari A, Bernard F, et al. i3DMM: deep implicit 3D morphable model of human heads. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 12803–12813
386 Wang X, Guo Y, Yang Z, et al. Prior-guided multi-view 3D head reconstruction. IEEE Trans Multimedia, 2021, 24: 4028–4040
387 Zheng Y, Abrevaya V F, Bühler M C, et al. IM avatar: implicit morphable head avatars from videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 13545–13555
388 Zhuang Y, Zhu H, Sun X, et al. MoFANeRF: morphable facial neural radiance field. In: Proceedings of European Conference on Computer Vision, 2022. 268–285
389 Hong Y, Peng B, Xiao H, et al. HeadNeRF: a real-time NeRF-based parametric head model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 20374–20384
390 Guo Y, Chen K, Liang S, et al. AD-NeRF: audio driven neural radiance fields for talking head synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 5784–5794
391 Gao X, Zhong C, Xiang J, et al. Reconstructing personalized semantic facial NeRF models from monocular video. ACM Trans Graph, 2022, 41: 1–12
392 Yin F, Zhang Y, Cun X, et al. StyleHEAT: one-shot high-resolution editable talking face generation via pre-trained styleGAN. In: Proceedings of European Conference on Computer Vision, 2022. 85–101
393 Wu W, Cao K, Li C, et al. Transgaga: geometry-aware unsupervised image-to-image translation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 8012–8021
394 Cudeiro D, Bolkart T, Laidlaw C, et al. Capture, learning, and synthesis of 3D speaking styles. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 10101–10111
395 Xing J, Xia M, Zhang Y, et al. CodeTalker: speech-driven 3D facial animation with discrete motion prior. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 12780–12790
396 Ye Z, Jiang Z, Ren Y, et al. GeneFace: generalized and high-fidelity audio-driven 3D talking face synthesis. In: Proceedings of the 11th International Conference on Learning Representations, 2023
397 Tang J, Wang K, Zhou H, et al. Real-time neural radiance talking portrait synthesis via audio-spatial decomposition. 2022. ArXiv:2211.12368
398 Aneja S, Thies J, Dai A, et al. ClipFace: text-guided editing of textured 3D morphable models. In: Proceedings of ACM SIGGRAPH 2023 Conference Proceedings, 2023. 1–11
399 Wang T, Zhang B, Zhang T, et al. Rodin: a generative model for sculpting 3D digital avatars using diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 4563–4573
400 Kolotouros N, Alldieck T, Zanfir A, et al. DreamHuman: animatable 3D avatars from text. In: Proceedings of Advances in Neural Information Processing Systems, 2024. 36
401 Alldieck T, Xu H, Sminchisescu C. imGHUM: implicit generative models of 3D human shape and articulated pose. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 5461–5470
402 Huang X, Shao R, Zhang Q, et al. HumanNorm: learning normal diffusion model for high-quality and realistic 3D human generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 4568–4577
403 Zhao M, Turner S J, Cai W. A data-driven crowd simulation model based on clustering and classification. In: Proceedings of IEEE/ACM 17th International Symposium on Distributed Simulation and Real Time Applications, 2013. 125–134
404 Wang S, Gain J E, Nistchke G S. Controlling crowd simulations using neuro-evolution. In: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, 2015. 353–360
405 Wei X, Lu W, Zhu L, et al. Learning motion rules from real data: neural network for crowd simulation. Neurocomputing, 2018, 310: 125–134
406 Lee J, Won J, Lee J. Crowd simulation by deep reinforcement learning. In: Proceedings of the 11th ACM SIGGRAPH Conference on Motion, Interaction and Games, 2018. 1–7
407 Hu K, Haworth B, Berseth G, et al. Heterogeneous crowd simulation using parametric reinforcement learning. IEEE Trans Visual Comput Graph, 2021, 29: 2036–2052
408 Panayiotou A, Kyriakou T, Lemonari M, et al. CCP: configurable crowd profiles. In: Proceedings of ACM SIGGRAPH 2022 Conference, 2022. 1–10
409 Charalambous P, Pettre J, Vassiliades V, et al. GREIL-crowds: crowd simulation with deep reinforcement learning and examples. ACM Trans Graph, 2023, 42: 1–15
410 Pataranutaporn P, Danry V, Leong J, et al. AI-generated characters for supporting personalized learning and well-being. Nat Mach Intell, 2021, 3: 1013–1022
411 Park J S, O'Brien J, Cai C J, et al. Generative agents: interactive simulacra of human behavior. In: Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, 2023. 1–22
412 Qian C, Liu W, Liu H, et al. Chatdev: communicative agents for software development. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, 2024. 15174–15186
413 Zhang H, Du W, Shan J, et al. Building cooperative embodied agents modularly with large language models. In: Proceedings of the 12th International Conference on Learning Representations, 2024
414 Lin J, Zhao H, Zhang A, et al. Agentsims: an open-source sandbox for large language model evaluation. 2023. ArXiv:2308.04026
415 Shinn N, Cassano F, Gopinath A, et al. Reflexion: language agents with verbal reinforcement learning. In: Proceedings of Advances in Neural Information Processing Systems, 2024. 36
416 Huang Z, Gutierrez S, Kamana H, et al. Memory sandbox: transparent and interactive memory management for conversational agents. In: Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, 2023. 1–3
417 Cambria E, Li Y, Xing F Z, et al. Senticnet 6: ensemble application of symbolic and subsymbolic AI for sentiment analysis. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020. 105–114
418 Zadeh A, Chen M, Poria S, et al. Tensor fusion network for multimodal sentiment analysis. 2017. ArXiv:1707.07250
419 Wei J, Wang X, Schuurmans D, et al. Chain-of-thought prompting elicits reasoning in large language models. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 35: 24824–24837
420 Wu W, Mao S, Zhang Y, et al. Mind's eye of LLMs: visualization-of-thought elicits spatial reasoning in large language models. In: Proceedings of the 38th Annual Conference on Neural Information Processing Systems, 2024
421 Lu Y, Yang S, Qian C, et al. Proactive agent: shifting LLM agents from reactive responses to active assistance. 2024. ArXiv:2410.12361
422 Deng Y, Liao L, Zheng Z, et al. Towards human-centered proactive conversational agents. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024. 807–818
423 Mueller F, Mehta D, Sotnychenko O, et al. Real-time hand tracking under occlusion from an egocentric RGB-D sensor. In: Proceedings

of the IEEE International Conference on Computer Vision, 2017. 1154–1163
424 Wang J, Mueller F, Bernard F, et al. Rgb2hands: real-time tracking of 3D hand interactions from monocular RGB video. ACM Trans Graph, 2020, 39: 1–16
425 Han S, Liu B, Cabezas R, et al. MEgATrack: monochrome egocentric articulated hand-tracking for virtual reality. ACM Trans Graph, 2020, 39: 1–13
426 Han S, Wu P C, Zhang Y, et al. UmeTrack: unified multi-view end-to-end hand tracking for VR. In: Proceedings of SIGGRAPH Asia 2022 Conference, 2022. 1–9
427 Streli P, Armani R, Cheng Y F, et al. HOOV: hand out-of-view tracking for proprioceptive interaction using inertial sensing. In: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, 2023. 1–16
428 Jiang X, Xiao Z G, Menon C. Virtual grasps recognition using fusion of leap motion and force myography. Virtual Reality, 2018, 22: 297–308
429 Diliberti N, Peng C, Kaufman C, et al. Real-time gesture recognition using 3D sensory data and a light convolutional neural network. In: Proceedings of the 27th ACM International Conference on Multimedia, 2019. 401–410
430 Arimatsu K, Mori H. Evaluation of machine learning techniques for hand pose estimation on handheld device with proximity sensor. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, 2020. 1–13
431 Chen T, Xu L, Xu X, et al. GestOnHMD: enabling gesture-based interaction on low-cost VR head-mounted display. IEEE Trans Visual Comput Graph, 2021, 27: 2597–2607
432 Lu C, Chakravarthula P, Tao Y, et al. Improved vergence and accommodation via Purkinje image tracking with multiple cameras for AR glasses. In: Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2020. 320–331
433 Wang Z, Zhao Y, Liu Y, et al. Edge-guided near-eye image analysis for head mounted displays. In: Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2021. 11–20
434 Xu M, Lu F. Gaze from origin: learning for generalized gaze estimation by embedding the gaze frontalization process. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2024. 38: 6333–6341
435 Liu R, Lu F. UVAGaze: unsupervised 1-to-2 views adaptation for gaze estimation. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2024. 38: 3693–3701
436 Hu Z, Li S, Zhang C, et al. DGaze: CNN-based gaze prediction in dynamic scenes. IEEE Trans Visual Comput Graph, 2020, 26: 1902–1911
437 Hu Z, Zhang C, Li S, et al. SGaze: a data-driven eye-head coordination model for realtime gaze prediction. IEEE Trans Visual Comput Graph, 2019, 25: 2002–2010
438 Stubbemann L, Dürrschnabel D, Refflinghaus R. Neural networks for semantic gaze analysis in XR settings. In: Proceedings of ACM Symposium on Eye Tracking Research and Applications, 2021. 1–11
439 Girado J, Peterka T, Kooima R, et al. Real time neural network-based face tracker for VR displays. In: Proceedings of IEEE Virtual Reality, 2007
440 Chen L, Cao C, Torre F D L, et al. High-fidelity face tracking for AR/VR via deep lighting adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 13059–13069
441 Teng T, Yang X. Facial expressions recognition based on convolutional neural networks for mobile virtual reality. In: Proceedings of the 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry, 2016. 475–478
442 Mo K, Guibas L J, Mukadam M, et al. Where2act: from pixels to actions for articulated 3D objects. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 6813–6823
443 Wang Y, Wu R, Mo K, et al. AdaAfford: learning to adapt manipulation affordance for 3D articulated objects via few-shot interactions. In: Proceedings of European Conference on Computer Vision, 2022. 90–107
444 Yang Y, Zhai W, Luo H, et al. LEMON: learning 3D human-object interaction relation from 2D images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 16284–16295
445 Yang Y, Zhai W, Wang C, et al. EgoChoir: capturing 3D human-object interaction regions from egocentric views. In: Proceedings of the 38th Annual Conference on Neural Information Processing Systems, 2024
446 Hong F, Guzov V, Kim H J, et al. EgoLM: multi-modal language model of egocentric motions. 2024. ArXiv:2409.18127
447 Sun G, Chen X, Chen Y, et al. Neural free-viewpoint performance rendering under complex human-object interactions. In: Proceedings of the 29th ACM International Conference on Multimedia, New York, 2021. 4651–4660
448 Wang Z, Chen Y, Jia B, et al. Move as you say, interact as you can: language-guided human motion generation with scene affordance. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024
449 Feng X, Bao Z, Wei S. Exploring CNN-based viewport prediction for live virtual reality streaming. In: Proceedings of IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR), 2019. 183–1833
450 Feng X, Liu Y, Wei S. LiveDeep: online viewport prediction for live virtual reality streaming using lifelong deep learning. In: Proceedings of IEEE Conference on Virtual Reality and 3D User Interfaces (VR), 2020. 800–808
451 Heyse J, Vega M T, de Backere F, et al. Contextual bandit learning-based viewport prediction for 360 video. In: Proceedings of IEEE Conference on Virtual Reality and 3D User Interfaces (VR), 2019. 972–973
452 Chen S, Duinkharjav B, Sun X, et al. Instant Reality: gaze-contingent perceptual optimization for 3D virtual reality streaming. IEEE Trans Visual Comput Graph, 2022, 28: 2157–2167
453 Khokhar A, Borst C W. Towards improving educational virtual reality by classifying distraction using deep learning. In: Proceedings of ICAT-EGVE, 2022. 85–90
454 Delvigne V, Wannous H, Vandeborre J P, et al. Attention estimation in virtual reality with EEG based image regression. In: Proceedings of IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR), 2020. 10–16
455 Li X, Shan Y, Chen W, et al. Predicting user visual attention in virtual reality with a deep learning model. Virtual Reality, 2021, 25: 1123–1136
456 Fathy F, Mansour Y, Sabry H, et al. Virtual reality and machine learning for predicting visual attention in a daylit exhibition space: a proof of concept. Ain Shams Eng J, 2023, 14: 102098
457 Strauss R R, Ramanujan R, Becker A, et al. A steering algorithm for redirected walking using reinforcement learning. IEEE Trans Visual Comput Graph, 2020, 26: 1955–1963
458 Chen Z Y, Li Y J, Wang M, et al. A reinforcement learning approach to redirected walking with passive haptic feedback. In: Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2021. 184–192
459 Wang M, Chen Z Y, Cai W C, et al. Transferable Virtual-Physical Environmental Alignment With Redirected Walking. IEEE Trans Visual Comput Graph, 2024, 30: 1696–1709
460 Lee D Y, Cho Y H, Lee I K. Real-time optimal planning for redirected walking using deep Q-learning. In: Proceedings of 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), 63–71
461 Jeon S B, Kwon S U, Hwang J Y, et al. Dynamic optimal space partitioning for redirected walking in multi-user environment. ACM Trans Graph, 2022, 41: 1–14
462 Azmandian M, Yahata R, Grechkin T, et al. Adaptive redirection: a context-aware redirected walking meta-strategy. IEEE Trans Visual Comput Graph, 2022, 28: 2277–2287
463 Cho Y H, Lee D Y, Lee I K. Path prediction using LSTM network for redirected walking. In: Proceedings of IEEE Conference on Virtual Reality and 3D User Interfaces (VR), 2018. 527–528
464 Duan T, Punpongsanon P, Iwai D, et al. FlyingHand: extending the range of haptic feedback on virtual hand using drone-based object

recognition. In: Proceedings of SIGGRAPH Asia 2018 Technical Briefs, 2018. 1–4
465 Yixian Y, Takashima K, Tang A, et al. ZoomWalls: dynamic walls that simulate haptic infrastructure for room-scale VR world. In: Proceedings of the 33rd annual ACM Symposium on User Interface Software and Technology, 2020. 223–235
466 Clarence A, Knibbe J, Cordeil M, et al. Unscripted retargeting: reach prediction for haptic retargeting in virtual reality. In: Proceedings of IEEE Virtual Reality and 3D User Interfaces (VR), 2021. 150–159
467 Salvato M, Heravi N, Okamura A M, et al. Predicting hand-object interaction for improved haptic feedback in mixed reality. IEEE Robot Autom Lett, 2022, 7: 3851–3857
468 Ge P, Pan J, Li F, et al. Real-time tracking of corneal contour in DALK surgical navigation using deep neural networks. In: Proceedings of IEEE International Conference on Image Processing (ICIP), 2019. 1356–1360
469 Alghofaili R, Sawahata Y, Huang H, et al. Lost in style: gaze-driven adaptive aid for VR navigation. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, 2019. 1–12
470 Seeliger A, Weibel R P, Feuerriegel S. Context-adaptive visual cues for safe navigation in augmented reality using machine learning. Int J Hum-Comput Interaction, 2024, 40: 761–781
471 Chen Z, Zeng W, Yang Z, et al. LassoNet: deep lasso-selection of 3D point clouds. IEEE Trans Visual Comput Graph, 2019, 26: 195–204
472 Cordeil M, Billy T, Mellado N, et al. Immersiveiml–immersive interactive machine learning for 3D point cloud classification: the neural network at your fingertips. In: Proceedings of IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct), 2023. 81–85
473 Himeur C E, Lejemble T, Pellegrini T, et al. PCEDNet: a lightweight neural network for fast and interactive edge detection in 3D point clouds. ACM Trans Graph, 2021, 41: 1–21
474 Marín-Morales J, Higuera-Trujillo J L, Greco A, et al. Affective computing in virtual reality: emotion recognition from brain and heartbeat dynamics using wearable sensors. Scientific Reports, 2018, 8: 13657
475 Gupta K, Lazarevic J, Pai Y S, et al. AffectivelyVR: towards VR personalized emotion recognition. In: Proceedings of the 26th ACM Symposium on Virtual Reality Software and Technology, 2020. 1–3
476 Šalkevicius J, Damaševičius R, Maskeliunas R, et al. Anxiety level recognition for virtual reality therapy system using physiological signals. Electronics, 2019, 8: 1039
477 Vaitheeshwari R, Yeh S C, Wu E H K, et al. Stress recognition based on multiphysiological data in high-pressure driving VR scene. IEEE Sens J, 2022, 22: 19897–19907
478 Martin N, Mathieu N, Pallamin N, et al. Virtual reality sickness detection: an approach based on physiological signals and machine learning. In: Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2020. 387–399
479 Liu M, Yang B, Xu M, et al. Exploring quantitative assessment of cybersickness in virtual reality using EEG signals and a CNN-ECA-LSTM network. Displays, 2024, 81: 102602
480 Dobre G C, Gillies M, Pan X. Immersive machine learning for social attitude detection in virtual reality narrative games. Virtual Reality, 2022, 26: 1519–1538
481 Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms. 2017. ArXiv:1707.06347
482 Zhao Y, Pan J, Dong Y, et al. Language urban odyssey: a serious game for enhancing second language acquisition through large language models. In: Proceedings of Extended Abstracts of the CHI Conference on Human Factors in Computing Systems, 2024. 1–7
483 Alghofaili R, Solah M S, Huang H, et al. Optimizing visual element placement via visual attention analysis. In: Proceedings of IEEE Conference on Virtual Reality and 3D User Interfaces (VR), 2019. 464–473
484 Horbova M, Andrunyk V, Chyrun L. Virtual reality platform using ML for teaching children with special needs. In: Proceedings of MoMLeT+ DS, 2020. 209–220
485 Tayal M A, Deshmukh M, Pangave V, et al. VMLHST: development of an efficient novel virtual reality ML framework with haptic feedbacks for improving sports training scenarios. Int J Electr Electron Res, 2023, 11: 601–608