Machine Learning-Based Denoising Techniques for Monte Carlo Rendering: A Literature Review

Liew Wen Yen[1], Rajermani Thinakaran[2], J. Somasekar[3]
Faculty of Data Science and Information Technology, INTI International University, Negeri Sembilan, Malaysia[1, 2]
Department of Computer Science and Engineering, Jain (Deemed–to-be University), Bangalore, Karnataka, India[3]

***Abstract*—Monte Carlo (MC) rendering is a powerful technique for achieving photorealistic images by simulating complex light interactions. However, the inherent noise introduced by MC rendering necessitates effective denoising techniques to enhance image quality. This paper presents a comprehensive review and comparative analysis of various machine learning (ML) methods for denoising MC renderings, focusing on four main categories: radiance prediction using convolutional neural networks (CNNs), kernel prediction networks, temporal rendering with recurrent architectures, and adaptive sampling approaches. Through systematic analysis of 7 peer-reviewed studies from 2019-2024, the author's findings reveal that deep learning models, particularly generative adversarial networks (GANs), achieve superior denoising performance. The study identifies key challenges including computational demands, with some methods requiring significant GPU resources, and generalization across diverse scenes. Additionally, we observe a trade-off between denoising quality and processing speed, particularly crucial for real-time applications. The study concludes with recommendations for future research, emphasizing the need for hybrid approaches combining physics-based models with ML techniques to improve robustness and efficiency in production environments.**

***Keywords*—Convolutional neural network; Monte Carlo rendering; generative adversarial network; deep learning; machine learning; denoising techniques**

I. Introduction

Monte Carlo (MC) rendering has emerged as a fundamental technique in computer graphics, enabling the simulation of light behavior in virtual environments through probabilistic sampling methods. By tracing numerous light paths and statistically sampling their contributions, MC rendering effectively captures complex light interactions with surfaces, materials, and volumes, resulting in highly realistic images characterized by accurate lighting, shadows, reflections, and refractions [1] [2]. This capability has rendered MC rendering indispensable across various industries, including film production, architectural visualization, and video game development, where photorealistic visuals are paramount [3] [4]. In film production, noise can disrupt the photorealism required for high-quality visual effects, while in video games, it can hinder real-time performance and user experience.

However, a notable challenge associated with MC rendering is the presence of noise in the generated images. Noise, which manifests as random variations or artifacts, arises from the inherent probabilistic nature of light path sampling. This issue is particularly pronounced in scenes with intricate lighting, glossy surfaces, or complex geometries, leading to grainy or speckled appearances that detract from the visual quality and realism of the rendered outputs [5] [6]. The reliance on probabilistic sampling methods contributes to this noise, as low sample counts can result in high variance in light estimates. While increasing the sample count can mitigate noise, it significantly escalates computational demands, rendering such approaches impractical for real-time or interactive applications [7].

To combat the noise prevalent in MC renderings, denoising techniques have become essential for enhancing image quality. These algorithms are designed to intelligently filter out noise while preserving critical image details, textures, and features, thereby yielding smoother and cleaner final renderings [7][8]. Recent advancements in machine learning (ML), particularly through deep learning models such as convolutional neural networks (CNNs) and generative adversarial networks (GANs), have opened new avenues for effective denoising. These ML techniques can learn complex noise patterns from extensive datasets of noisy and clean images, enabling them to generalize across various scenes and lighting conditions while maintaining the fidelity of essential image details during the denoising process [3] [9].

Despite the extensive exploration of ML-based denoising methods, there remains a scarcity of work synthesizing and comparing these approaches across diverse rendering scenarios. Challenges such as high computational demands, generalization across varying scenes, and interpretability of the models persist as unresolved issues. High computational demands limit the applicability of ML-based denoising in real-time applications, while generalization issues arise due to the variability of noise patterns across different scenes and lighting conditions. This study endeavors to fill these gaps by providing a systematic review and comparative analysis of existing methods, offering practical recommendations for future research in the field of MC rendering and denoising [5] [9].

To address these challenges, this study seeks to answer the following key research questions:

1) What ML methods have been employed for denoising MC renderings, according to the existing literature?

2) How do these ML methods compare in terms of performance, efficiency, and application?

3) What are the current challenges and future directions for ML-based denoising in MC rendering?

The primary objective of this study is to provide a comprehensive review and comparative analysis of these techniques, categorizing them into radiance prediction, kernel prediction, temporal rendering, and adaptive sampling. Performance evaluations will utilize metrics such as Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Relative Mean Squared Error (rMSE) [4] [8]. Additionally, this study will highlight gaps in the current literature and propose future research directions to advance ML-based denoising techniques, ultimately enhancing rendering workflows and visual outputs.

II. METHOD

A systematic literature review (SLR) methodology was employed to ensure a comprehensive and unbiased review of ML techniques for denoising in MC rendering. An SLR involves analyzing existing research by defining clear research questions, identifying relevant studies, appraising their quality, and synthesizing findings both qualitatively and quantitatively [10]. This structured approach ensures transparency, replicability, and rigor in the review process. The methodology adheres to the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) framework to enhance methodological rigor and quality.

The methodology of this study is organized into three key stages: planning the review, which involves defining research questions, developing search strategies, and establishing inclusion/exclusion criteria; conducting the review, which includes searching and screening relevant studies, extracting data, and assessing the quality of selected studies; and analyzing the gathered information, which consists of synthesizing results, discussing trends, and identifying challenges and opportunities for future research.

A. Planning the Review

1) Scope of the review: The SLR focuses specifically on machine learning ML applied to denoising in MC rendering. MC rendering is widely used in computer graphics to simulate realistic lighting effects, but its probabilistic nature often introduces noise into rendered images. This noise can degrade visual quality, making effective denoising techniques essential for achieving high-quality outputs. Despite recent advancements in ML-based denoising methods, there remains a scarcity of work synthesizing and comparing these approaches across diverse rendering scenarios. Challenges such as high computational demands, generalization across varying scenes, and interpretability of the models persist as unresolved issues. This study aims to address these gaps by providing a comprehensive review and comparative analysis of ML-based denoising techniques, categorizing them into radiance prediction, kernel prediction, temporal rendering, and adaptive sampling.

2) Research questions: The formulation of research questions was guided by an iterative process involving pilot searches and consultations with domain experts. Initial exploratory searches were conducted across academic databases such as IEEE Xplore, ACM Transactions on Graphics, and ScienceDirect using broad keywords like “Monte Carlo rendering,” “denoising,” and “machine learning.” These searches helped identify recurring themes and trends in literature, such as the use of CNNs, GANs, and kernel prediction methods. Informal consultations with an expert in computer graphics and ML-based rendering techniques provided valuable feedback on the scope and relevance of the questions, suggesting additional considerations such as computational efficiency and generalization across diverse scenes. The final research questions guiding this SLR are as follows:

1) What ML methods have been employed for denoising MC renderings, according to the existing literature?

2) How do these ML methods compare in terms of performance, efficiency, and application?

3) What are the current challenges of ML for denoising MC rendering?

4) Search strategy: A comprehensive search strategy was developed to gather relevant literature across multiple academic databases. The search was conducted using keywords such as “Monte Carlo rendering,” “denoising,” “convolutional neural network,” “deep learning,” “machine learning denoising techniques,” and “generative adversarial network”. The databases searched included IEEE Xplore, Google Scholar, ScienceDirect, Computer Graphics Forum, and ACM Transactions on Graphics. To ensure consistency in the analysis, the search was limited to studies published in English between 2019 and 2024.

5) Inclusion and exclusion criteria: Specific criteria were established to filter the studies for inclusion in this review, ensuring the relevance and quality of the selected literature. This study employed a filtration process guided by predefined inclusion and exclusion criteria as shown in Tables I and II. All the papers were assessed against a set of inclusion and exclusion criteria to ensure they directly addressed the research questions.

TABLE I. INCLUSION CRITERIA

ID	Inclusion criterion
I1	Studies explicitly focus on machine learning techniques for denoising in Monte Carlo rendering.
I2	Studies published in peer-reviewed journals and conferences to ensure credibility and quality.
I3	Studies that provide quantitative performance metrics such as PSNR, SSIM, rMSE, or computational efficiency.
I4	Studies published in English.
I5	Studies published between 2019 and 2024.

TABLE II. EXCLUSION CRITERIA

ID	Exclusion criterion
E1	Studies do not focus on denoising in the context of Monte Carlo rendering.
E2	Studies lacking sufficient empirical data or clear evaluation methods, which could undermine the validity of the findings.
E3	Non-peer-reviewed articles, editorials, or opinion pieces, as these sources do not provide the rigorous analysis required for this review.
E4	Studies published in languages other than English.
E5	Studies published before 2019.

6) Selection process: The selection process involved two phases: initial screening and full-text review. In the initial screening, titles and abstracts of all retrieved papers were reviewed against the inclusion and exclusion criteria. Papers that did not meet these criteria were excluded. In the full-text review, the remaining papers were examined in detail to assess their relevance to the research questions and the quality of their methodologies and findings. A total of seven papers were selected for the final analysis. To provide a clear overview of the study selection process, a PRISMA flowchart (Fig. 1) was created, following the guidelines outlined by Pati and Lorusso [11]. The flowchart visually summarizes the number of studies identified, screened, and included at each stage of the review process.

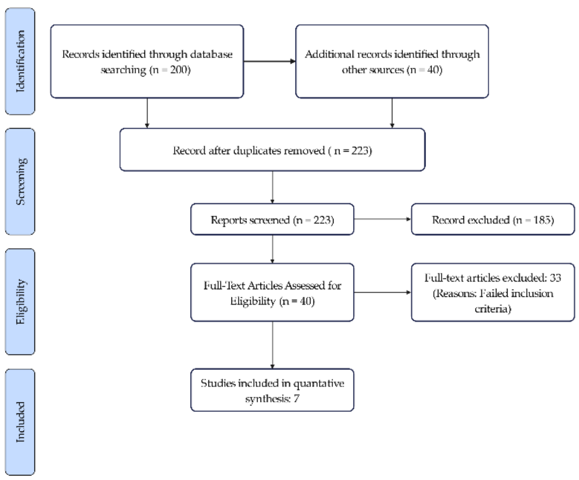

Fig. 1. PRISMA flowchart summarizing the study selection process.

B. Conducting the Review

This phase involved three key steps: data extraction, quality assessment, and data synthesis. Each step was meticulously performed to ensure a robust and unbiased evaluation of the selected studies, enabling a comprehensive comparison of ML-based denoising techniques for MC rendering.

1) Data extraction: Data extraction was systematically performed on the selected studies to comprehensively address the research questions. Key information extracted included the specific ML models employed for denoising, the datasets used, and performance metrics such as PSNR and SSIM, rMSE. This structured extraction approach enabled a robust comparison of the effectiveness and efficiency of different ML-based denoising techniques across a range of rendering scenarios.

2) Quality assessment: To ensure the reliability and validity of the findings, each study included in this review underwent a rigorous quality assessment. This process evaluated the methodological rigor of the studies, focusing on factors such as the robustness of the experimental design, the clarity of data presentation, and the appropriateness of the performance metrics used. Only studies that met these stringent criteria were included, ensuring that this review comprises high-quality research offering credible insights into the effectiveness and efficiency of ML-based denoising methods.

3) Data synthesis: The extracted data were synthesized to provide a comprehensive comparison of the different ML-based denoising techniques for MC rendering. Both qualitative and quantitative analyses were conducted to identify trends, strengths, and limitations across the existing literature. Performance metrics were aggregated where applicable, allowing for a standardized comparison of the denoising effectiveness across different studies. This synthesis provides a holistic view of the current landscape of ML-based denoising methods in MC rendering, highlighting their practical applications and potential areas for future research.

C. Analyzing the Gathered Information

1) Synthesis of results: The extracted data were synthesized to provide a comprehensive comparison of the different ML-based denoising techniques for MC rendering. Both qualitative and quantitative analyses were conducted to identify trends, strengths, and limitations across the existing literature. Performance metrics were aggregated where applicable, allowing for a standardized comparison of the denoising effectiveness across different studies. This synthesis provides a holistic view of the current landscape of ML-based denoising methods in MC rendering, highlighting their practical applications and potential areas for future research.

2) Discussion of results: The analysis identified several challenges associated with applying ML techniques to MC rendering denoising. One major challenge is the computational complexity of these methods, as many ML-based denoisers require substantial processing power and memory to achieve high-quality results. This computational demand makes it difficult to deploy these techniques in real-time or interactive applications where performance and speed are critical. Additionally, the complexity of the models can hinder their ability to generalize across diverse scenes, as training datasets may not fully capture the variability in noise patterns that arise in different rendering scenarios.

Another challenge is the difficulty in balancing denoising performance with computational efficiency. While deep learning models such as CNNs and GANs have shown promise in reducing noise while preserving image details, these methods often come with a trade-off between the quality of the denoised output and the computational resources required. Addressing these challenges involves exploring more efficient architectures, optimization techniques, and potentially new approaches to model training that can reduce computational overhead without compromising denoising quality.

3) Recommendations for future research: Based on the findings, several recommendations for future research have been proposed. These include integrating physics-based models, adopting adaptive sampling strategies, employing advanced network architectures such as GANs and CNNs, utilizing detail-preserving neural networks, and implementing path-based denoising techniques. These suggestions aim to steer future research efforts toward overcoming current limitations and exploring new opportunities in ML-based denoising for MC rendering.

III. Results and Discussion

This study explores three key questions. The following sections analyze the findings and their significance to each question.

A. *What ML Methods have been Employed for Denoising MC Rendering, according to the Existing Literature?*

MC rendering is well-known for simulating realistic lighting effects, but it has its challenges, particularly with noise in the images because of the stochastic nature of the sampling process. Over the years with the advancement of ML, it has been employed to effectively denoise MC renderings. These methods leverage neural networks which enables them to clean up images more effectively than traditional techniques. Below, we explore some of the research papers that use ML methods for denoising MC rendering, categorized into kernel prediction, parameter prediction, radiance prediction, and temporal denoising. This categorization framework, derived from the work of Huo et al. [12], provides a structured approach to understanding the strengths and limitations of each method.

1) Kernel prediction: Kernel prediction focuses on directly predicting the filtering kernels used to combine neighboring pixel values. This enhances the denoising process by adapting the kernels to the specific noise characteristics of each pixel. This approach is a more flexible and accurate solution compared to traditional filtering techniques, particularly in handling complex scenes and varying lighting conditions.

Back et al. [13] introduce a deep learning-based framework designed to improve the accuracy of MC rendering by effectively combining independent and correlated pixel estimates. Their approach utilizes a combination kernel modeled as a deep neural network, which optimally weights the combination of these pixel estimates, thereby reducing residual noise and systematic errors commonly found in existing methods like denoising and gradient-domain rendering. The framework is robust against outliers, thanks to an extension that employs multi-buffered inputs, which further enhances the reliability of the results. Experimental evaluations demonstrate that this method not only enhances the visual quality of renders by preserving high-frequency details and reducing noise but also outperforms existing techniques in terms of both numerical accuracy and visual fidelity. This makes the approach particularly valuable for applications requiring high-quality rendering, such as production-level visual effects and interactive applications.

Gharbi et al. [14] introduce a sample-based MC denoising technique using a kernel-splatting network. Unlike traditional pixel-based methods, their approach operates directly on the raw MC samples, leveraging deep learning to map these samples to a denoised image. The core innovation lies in a novel kernel-predicting architecture that splats individual samples onto nearby pixels. This method treats each sample independently and uses a permutation-invariant design to handle the arbitrary order of samples. The kernel-splatting approach is particularly effective in managing complex light transport scenarios such as motion blur, depth of field, and specular effects. By directly processing the sample-level information, the technique achieves higher quality results with reduced numerical error and improved visual fidelity, especially in low-sample-count settings. The network was trained on a large dataset of synthetic scenes and demonstrated significant improvements over state-of-the-art methods in both visual quality and computational efficiency [14].

Munkberg and Hasselgren et al. [15] propose a novel approach to neural denoising in MC path tracing by introducing a layered architecture that partitions per-sample data into distinct layers. Each layer is processed with unique filter kernels before being composited to produce the final output. This approach balances computational efficiency with high-quality denoising, offering comparable results to more expensive per-sample methods while significantly reducing memory and performance overhead. The architecture is particularly robust against high-intensity outliers and performs well even in complex visibility scenarios, such as defocus and motion blur. The authors demonstrate that their method achieves near real-time performance on contemporary GPUs, making it viable for both real-time rendering and offline production environments. Future work is suggested in extending this layered approach to temporal domains and deep compositing workflows, indicating its potential for broader applications in rendering technologies.

2) Parameter prediction: Parameter prediction involves training neural networks to predict the optimal parameters for traditional filters to enhance their ability to reduce noise while preserving image details.

Xing and Chen [16] introduce an approach to denoising path-traced images by combining SURE-based adaptive sampling with neural networks. Their process begins with generating coarse samples and using Stein's Unbiased Risk Estimator (SURE) to estimate the noise level for each pixel. Extra samples are then allocated to pixels with higher noise levels. In the reconstruction phase, a MLP network predicts the optimal reconstruction parameters based on features extracted from the adaptive sampling results, such as shading normal, depth, and texture values. These predicted parameters are used with an anisotropic filter to produce the final noise-free image. This method reduces numerical error as well as enhances visual quality compared to existing techniques.

3) Radiance prediction: Radiance prediction focuses on directly estimating the radiance values for each pixel in a MC rendering. These methods bypass the need for traditional filtering or kernel prediction. They utilize deep learning models to map noisy input pixels directly to their denoised counterparts, effectively capturing complex relationships between the noisy input and the desired output. By predicting radiance directly, these approaches can handle high-frequency details and complex lighting scenarios more effectively, allowing them to be more suitable for applications where visual accuracy is needed.

Xu et al. [17] introduce an adversarial approach for denoising MC renderings, leveraging GANs to improve the realism of high-frequency details and global illumination. Their method employs a conditioned auxiliary feature modulation technique that utilizes auxiliary features such as normal, albedo, and depth to enhance the denoising process. The GAN framework consists of a denoising network, which predicts the clean image, and a critic network, which evaluates the perceptual quality of the denoised output. The critic network is trained using the Wasserstein distance, which provides a smoother measure of perceptual similarity compared to traditional losses. This approach enables the denoising network to learn from the distribution of high-quality path-traced images, resulting in better reconstruction of MC integrals from fewer samples. Xu et al. demonstrate that their method outperforms previous state-of-the-art techniques in terms of both visual quality and computational efficiency, making it suitable for high-end production environments.

The paper by Alsaiari et al. [18] presents a novel approach for image denoising using GAN architecture. The method involves rendering images with a reduced number of samples per pixel, which results in noisy outputs, and then passing these images through a GAN-based network that produces high-quality, photorealistic denoised images in less than a second. The proposed network architecture leverages residual blocks, skip connections, and batch normalization to enhance the denoising process. Despite being trained on a limited dataset of 40 images, the network demonstrated impressive generalization capabilities, effectively denoising images outside the training domain, including grainy photographs and medical CT scans. The authors also discuss potential future extensions of their work, including handling more complex noise patterns such as those generated by MC rendering and incorporating additional information like depth maps to improve denoising performance in scenes with motion blur, depth of field, and global illumination. The study underscores the effectiveness of GANs in producing high-quality denoised images and suggests further exploration of this approach in real-time rendering applications.

4) Temporal rendering: Temporal rendering is specifically designed to address the challenges of ensuring frame-to-frame consistency in animated or real-time rendering sequences. Noise reduction in MC rendering needs to be effective not just on individual frames, but also across time, to prevent flickering or temporal artifacts that can detract from the visual experience. These methods often utilize recurrent structures to ensure that noise is reduced consistently across frames. It preserves temporal coherence while maintaining high-quality image details.

Meng et al. [19] introduce a practical and efficient approach to real-time MC denoising by leveraging a neural bilateral grid. Their method utilizes a convolutional neural network, called GuideNet, to predict guide images that direct the placement of noisy radiance data into a multi-scale bilateral grid. The grid is then sliced to extract denoised data, resulting in high-quality renders even from extremely noisy inputs at 1 spp. The proposed approach is highly scalable and adaptable to both real-time and offline applications, demonstrating superior denoising quality compared to existing methods, particularly for low-sample scenarios. The study emphasizes the method's ability to maintain interactive frame rates while achieving high visual fidelity, making it a robust solution for real-time rendering in demanding environments.

B. How do these ML Methods Compare in Terms of Performance, Efficiency, and Application?

These denoising methods analyzed in this study exhibit varying degrees of performance, efficiency, and application suitability in MC-rendered images. By examining key metrics such as rMSE, SSIM, PSNR, and processing time, we can assess how each method balances noise reduction, computational efficiency, and applicability to different rendering scenarios.

Methods such as Xu et al. [17] excel with an rMSE of 0.003164 and a PSNR of 34.194759 dB in the HorseRoom scene, outperforming traditional methods like NFOR in retaining fine details. These methods are particularly suited for scenarios where achieving the highest possible image quality is crucial, even if it comes at the cost of longer processing times.

In environments where real-time performance is essential, such as video games, virtual reality, and interactive simulations, the kernel-splatting network by Gharbi et al. [14] and the GAN-based approach by Alsaiari et al. [18] are particularly effective. Gharbi et al.'s method [14] achieves an rMSE of 0.026 at 32 spp while processing a 1024 × 1024 image in just 6.0 seconds at 4 spp, making it ideal for real-time applications that require a balance between speed and quality. Similarly, Alsaiari et al.'s method [18] generates high-quality denoised images in under a second, emphasizing rapid processing without significantly compromising visual fidelity, making it highly suitable for scenarios where quick turnaround times are critical.

Methods by Jonghee Back et al. [13] and Munkberg and Hasselgren [15] offer strong capabilities in handling complex lighting environments and preserving intricate details. Jonghee Back et al.'s deep combiner for independent and correlated pixel estimates achieves a significant reduction in rMSE, such as 0.0207 in the Bookshelf scene at 64 spp, making it effective in handling scenes with intricate lighting and textures. Munkberg and Hasselgren's neural denoising method with layer embedding also shows strong performance, achieving an rMSE of 0.0288 and SSIM of 0.941 at 32 spp, making it highly effective for maintaining image quality in complex visual effects.

Xing and Chen's method [16] leverages adaptive sampling based on SURE combined with a modified MLP network to predict optimal reconstruction parameters. The method demonstrates significant noise reduction with a RelMSE of 0.00831 in the Sibenik scene at 16.9 spp, and 2.37E-4 in the Anim-BlueSphere scene at 30.6 spp. It optimizes sample distribution across pixels with varying noise levels, enhancing computational efficiency while maintaining high image quality. This method is particularly effective for real-time applications and interactive graphics, where maintaining quality with lower sample counts is crucial.

C. What are the Current Challenges in the Application of ML for Denoising MC Rendering?

One of the main challenges associated with using ML for denoising in MC rendering is its inherent complexity. MC rendering simulates how light behaves within a scene by tracing numerous random paths. Hence, this results in inherently noisy images and requires a denoising process for better visual quality [20]. The primary difficulty lies in the nature of the noise, which is stochastic and can vary significantly across different scenes. Therefore, denoising algorithms must be adept at distinguishing between noise and the true signal to avoid blurring or distorting the final image [20]. In addition, the denoising task is complicated both by the high dimensionality of the data and the complex interplay of light transport phenomena. In this regard, a sophisticated ML model is needed to be able to capture these intricate relationships with complex data [21, 22].

Additionally, the efficiency and computational cost of denoising algorithms are also challenges in the MC rendering process. Deep learning techniques have shown promise in attaining higher quality denoising, but they often come with a computational cost and require substantial processing power and time for training and inference [23]. This computational overhead can make deploying ML-based denoising solutions in real-time or interactive rendering scenarios challenging, where performance is crucial [24]. Real-time rendering applications, such as video games or virtual reality, demand quick and efficient processing to maintain smooth and responsive user experiences. Therefore, balancing denoising quality and computational efficiency in rendering and MC is today's critical challenge using ML approaches for denoising [23]. Developing methods that optimize this balance is essential to ensure that high-quality denoising can be achieved without compromising the performance required for real-time applications. This involves exploring more efficient algorithms, hardware acceleration, and innovative training techniques to reduce computational demands while maintaining or improving denoising effectiveness.

The other critical challenge is the generalization of denoising algorithms across different scenes and lighting conditions. Noise patterns and characteristics of MC renderings can vary greatly depending on scene complexity, materials present, and lighting setup [25]. For instance, a scene with complex geometry and reflective surfaces might produce noise patterns that are vastly different from a simple scene with diffuse materials. This variability necessitates that ML models are not only trained on diverse datasets but are also rigorously tested to ensure their effectiveness in new, unseen scenarios. Thus, it is essential to practically assess that ML models effectively generalize over unseen data and across diverse rendering scenarios in rendering pipelines [25]. Robustness to scene variations and the ability to adapt to different noise profiles are crucial aspects that need to be addressed to make denoising algorithms effective across a wide range of rendering scenarios [25][26][27]. Addressing these issues involves developing more sophisticated training regimes, incorporating a wider range of scenes and conditions, and continuously updating models to handle new types of noise as they are encountered.

Besides, ML-based denoising methods further raise issues concerning interpretability and transparency for MC rendering. Deep learning models are often thought to be black boxes, making it difficult to understand the decision process for denoising and what features were prioritized in the process [21]. This can be problematic for artists and developers who rely on precise control over rendering parameters to achieve specific visual effects [21]. For instance, they may need to adjust the denoising parameters to maintain certain artistic details or to ensure the consistency of visual styles across different scenes. Without a clear understanding of how the ML model operates, making these adjustments becomes exceedingly difficult. This lack of interpretability can also hinder debugging and improvement efforts, as it is unclear why the model might fail in certain scenarios. Thus, improving the interpretability of these models while preserving their denoising performance is a challenge that needs attention in applying ML for denoising MC rendering [21].

Table III provides a summary of the performance, efficiency, and application of a selection of seven methods. This table serves as a quick reference for understanding the strengths and limitations of each approach, making it easier for researchers and practitioners to select the most appropriate denoising method based on their specific needs. By comparing metrics such as PSNR, SSIM, and computational demands, the table illustrates the diverse range of strategies employed across different methods to balance speed and quality.

IV. CONCLUSION

This study set out to explore and analyze ML-based denoising techniques for MC rendering, focusing on three key research questions. Using a SLR approach guided by the PRISMA framework, the authors identified, categorized, and compared seven peer-reviewed studies published between 2019 and 2024. These methods were grouped into four main categories—radiance prediction, kernel prediction, temporal rendering, and adaptive sampling—and evaluated using metrics like PSNR, SSIM, and rMSE. Our findings show that deep learning models, such as CNNs and GANs, are highly effective at reducing noise while preserving important details in MC-rendered images. However, challenges like high computational demands and difficulties in generalizing across different scenes still limit their use in real-time applications.

The main contribution of this work is bringing together a fragmented field into a clear and structured framework. This makes it easier for practitioners to choose the right denoising technique based on their specific needs. For example, kernel-splatting networks work well for real-time scenarios, while adversarial methods excel in producing high-quality results for offline production. We also highlighted some critical gaps in the literature, such as the need for more interpretable models and efficient architectures that strike a better balance between quality and computational cost.

That said, this review isn't without its limitations. By focusing only on studies from 2019 to 2024, we might have missed some foundational work from earlier years. Additionally, our reliance on databases like IEEE Xplore, ACM, and ScienceDirect could introduce bias, and the inclusion of just 12 papers may not fully capture the diversity of approaches out there. While qualitative comparisons provide valuable insights, they lack the statistical depth of a meta-analysis, which could offer a more quantitative assessment of these methods.

Looking ahead, there are several exciting directions for future research. One promising area is hybrid approaches that combine physics-based models with ML techniques to improve robustness and accuracy. Another is exploring temporal optimization strategies to reduce flickering artifacts in animations. By addressing these challenges, researchers can develop deployable solutions that balance photorealism with computational efficiency, ultimately transforming workflows in industries like film, architecture, and gaming.

In short, this study provides a comprehensive overview of the current state of ML-based denoising for MC rendering, identifies key challenges, and suggests practical ways forward. The goal is to inspire further innovation in creating denoising solutions that are not only powerful but also practical for real-world applications.

TABLE III. SUMMARY OF THE PERFORMANCE, EFFICIENCY, AND APPLICATION OF THE METHODS

Method	rMSE (Range)	SSIM (Range)	PSNR (dB)	Efficiency	Application
Adversarial Monte Carlo Denoising by Xu et al. [17]	0.003164	N/A	34.194759	High computational cost, suited for offline rendering	High-end production environments, detailed visual effects
Kernel-Splatting Network by Gharbi et al. [14]	0.026	N/A	N/A	Optimized for fewer samples, balance between speed and quality	Real-time applications, interactive graphics, gaming
GAN-based Denoising by Alsaiari et al.[18]	N/A	0.938 - 0.941	33.706 - 33.878	Real-time performance, quick processing in under a second	Real-time rendering, architectural visualization
Neural Bilateral Grid by Meng et al. [19]	N/A	0.941	33.838	High real-time performance at 61 FPS, optimized for low sample scenarios	Real-time rendering, gaming, and VR
Deep Combiner for Independent and Correlated Pixel Estimates by Jonghee Back et al. [13]	N/A	N/A	N/A	Effective in handling complex scenes, reduces relMSE	Production-level visual effects, interactive applications requiring high-quality rendering
Neural Denoising with Layer Embedding by Munkberg and Hasselgren [15]	0.0288	0.941	N/A	Robust against artifacts, effective with varying configurations	Offline rendering, flexible in handling per-sample, per-pixel, and layered configurations
Path Tracing Denoising based on SURE Adaptive Sampling by Xing and Chen [16]	0.00831	N/A	N/A	Adaptive sampling with SURE; highly efficient in real-time scenarios with CUDA acceleration	Interactive graphics, real-time applications, scenarios with limited computational resources

REFERENCES

[1] K. Wong and T. Wong, "Deep residual learning for denoising monte carlo renderings", Computational Visual Media, vol. 5, no. 3, p. 239-255, 2019.

[2] J. Back, B. Hua, T. Hachisuka, & B. Moon, "Deep combiner for independent and correlated pixel estimates", Acm Transactions on Graphics, vol. 39, no. 6, p. 1-12, 2020.

[3] Q. Hou, "Auxiliary features-guided super resolution for monte carlo rendering", Computer Graphics Forum, vol. 43, no. 1, 2023.

[4] X. Zhang, M. Manzi, T. Vogels, H. Dahlberg, M. Groß, & M. Papas, "Deep compositional denoising for high-quality monte carlo rendering", Computer Graphics Forum, vol. 40, no. 4, p. 1-13, 2021.

[5] C. Zhang, D. Zhang, M. Doggett, & S. Zhao, "Antithetic sampling for monte carlo differentiable rendering", Acm Transactions on Graphics, vol. 40, no. 4, p. 1-12, 2021.

[6] W. Lin, B. Wang, L. Wang, & N. Holzschuch, "A detail preserving neural network model for monte carlo denoising", Computational Visual Media, vol. 6, no. 2, p. 157-168, 2020.

[7] B. Hua, A. Gruson, V. Petitjean, M. Zwicker, D. Nowrouzezahrai, E. Eisemannet al., "A survey on gradient-domain rendering", Computer Graphics Forum, vol. 38, no. 2, p. 455-472, 2019.

[8] A. Firmino, J. Frisvad, & H. Jensen, "Progressive denoising of monte carlo rendered images", Computer Graphics Forum, vol. 41, no. 2, p. 1-11, 2022.

[9] J. Lee, "Real-time monte carlo denoising with adaptive fusion network", Ieee Access, vol. 12, p. 29154-29165, 2024.

[10] H. Dahlberg, D. Adler, & J. Newlin, "Machine-learning denoising in feature film production", ACM SIGGRAPH 2019 Talks, 2019.

[11] D. Pati and L. N. Lorusso, "How to Write a Systematic Review of the Literature," HERD: Health Environments Research & Design Journal, vol. 11, no. 1, pp. 15–30, Dec. 2018.

[12] Y. Huo and S. Yoon, "A survey on deep learning-based Monte Carlo denoising," Computational Visual Media, vol. 7, no. 2, pp. 169–185, Mar. 2021.

[13] J. Back, B. Hua, T. Hachisuka, & B. Moon, "Deep combiner for independent and correlated pixel estimates", ACM Transactions on Graphics, vol. 39, no. 6, p. 1-12, 2020.

[14] M. Gharbi, T. Li, M. Aittala, J. Lehtinen, & F. Durand, "Sample-based monte carlo denoising using a kernel-splatting network", ACM Transactions on Graphics, vol. 38, no. 4, p. 1-12, 2019.

[15] J. Hasselgren, J. Munkberg, M. Salvi, A. Patney, & A. Lefohn, "Neural temporal adaptive sampling and denoising", Computer Graphics Forum, vol. 39, no. 2, p. 147-155, 2020.

[16] Q. Xing and C. Chen, "Path Tracing Denoising Based on SURE Adaptive Sampling and Neural Network," IEEE Access, vol. 8, pp. 116336–116349, Jan. 2020.

[17] B. Xu, J. Zhang, R. Wang, K. Xu, Y. Yang, C. Liet al., "Adversarial monte carlo denoising with conditioned auxiliary feature modulation", ACM Transactions on Graphics, vol. 38, no. 6, p. 1-12, 2019.

[18] A. Alsaiari, R. Rustagi, A. Alhakamy, M. M. Thomas and A. G. Forbes, "Image Denoising Using A Generative Adversarial Network," 2019 IEEE 2nd International Conference on Information and Computer Technologies (ICICT), Kahului, HI, USA, 2019, pp. 126-132.

[19] X. Meng, Q. Zheng, A. Varshney, G. Singh, and M. Zwicker, "Real-time Monte Carlo Denoising with the Neural Bilateral Grid," Eurographics, pp. 13–24, Jan. 2020.

[20] A. Kuznetsov, N. K. Kalantari, and R. Ramamoorthi, "Deep Adaptive Sampling for Low Sample Count Rendering," Computer Graphics Forum, vol. 37, no. 4, pp. 35–44, Jul. 2018.

[21] X. Zhang, M. Manzi, T. Vogels, H. Dahlberg, M. Groß, & M. Papas, "Deep compositional denoising for high-quality monte carlo rendering", Computer Graphics Forum, vol. 40, no. 4, p. 1-13, 2021.

[22] K. Wong and T. Wong, "Deep residual learning for denoising monte carlo renderings", Computational Visual Media, vol. 5, no. 3, p. 239-255, 2019.

[23] X. Yang, D. Wang, W. Hu, L. Zhao, X. Piao, D. Zhouet al., "Fast reconstruction for monte carlo rendering using deep convolutional networks", Ieee Access, vol. 7, p. 21177-21187, 2019.

[24] C. R. A. Chaitanya et al., "Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder," ACM Transactions on Graphics, vol. 36, no. 4, pp. 1–12, Jul. 2017.

[25] W. Lin, B. Wang, L. Wang, & N. Holzschuch, "A detail preserving neural network model for monte carlo denoising", Computational Visual Media, vol. 6, no. 2, p. 157-168, 2020. https://doi.org/10.1007/s41095-020-0167-7R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[26] M. Boughida and T. Boubekeur, "Bayesian collaborative denoising for monte carlo rendering", Computer Graphics Forum, vol. 36, no. 4, p. 137-153, 2017.

[27] D. Sukmawan, D.O.D Handayani, and D. A. Dewi, "Deep Learning Approaches to Identify Sukabumi Potentials Through Images on Instagram", In 2021 IEEE 7th International Conference on Computing, Engineering and Design (ICCED) (pp. 1-6). IEEE, 2021.