NeRFix: fixing subtle mistakes in the quadrature of the NeRF volumetric integral

Luca Savant Aira
Politecnico di Torino

Diego Valsesia
Politecnico di Torino

Enrico Magli
Politecnico di Torino

Abstract

Neural Radiance Fields (NeRFs) are popular models using MLPs for continuous volumetric representations, which render novel views by means of numerical quadrature of the volumetric integral. Despite the widespread usage of NeRFs, we report subtle mistakes in their quadrature rule which have gone unnoticed for many years. In particular, we highlight mistakes present in both the original NeRF paper and its implementation as well as in the mainstream NeRFStudio implementation. We provide a theoretical analysis of rendering error bounds for NeRF and we use it to show suboptimal convergence rates in the presence of the aforementioned mistakes. We introduce minimal fixes that recover the expected convergence orders with no complexity overhead. Finally, while fixing the quadrature scheme is mostly a matter of theoretical correctness, we also provide experimental evidence that modest improvements in image quality are achieved on standard benchmarks.

1 Introduction

Novel view synthesis seeks to generate photorealistic images of a scene from previously unseen viewpoints. Neural Radiance Fields (NeRFs)[15] have become a central approach to this problem by representing with multi-layer perceptrons (MLPs) a continuous radiance field of a scene and rendering pixels through the volumetric integral (VI) along camera rays. The seminal NeRF paper has spurred a large number of methods, and related software frameworks, focusing on specific aspects of the problem: operating on unposed images [11, 25, 5, 10], handling appearance variation or exposure changes [12, 16, 7], processing dynamic scenes [20, 19], improving network architecture [2, 3, 17, 21], including regularization priors [4, 9, 18].

To the best of our knowledge, there is a gap in the literature, as prior NeRF works did not analyze the convergence order of the VI quadrature schemes. In this paper, we analytically study the quadrature rules used to numerically compute the VI in two widely used formulations – the original NeRF paper [15] and the commonly used `SpacedSampler` path in NeRFStudio [23]– and we report subtle mistakes present in them and, as a cascade effect, in the vast majority of papers deriving their implementations from them. In particular, the original NeRF paper employs an estimator that, read as written, is not well-defined, and its official implementation does not ultimately fix it. On the other hand, the analyzed NeRFStudio sampler avoids this particular inconsistency, but does so by effectively changing the VI integration domain, thereby introducing another error. These mistakes are easy to miss in practice because training still proceeds and the model still produces reasonable images. However, even on closed-form constant-color test cases, these schemes exhibit at-most first-order convergence behavior (or no convergence at all), whereas proper schemes implemented by simple fixes recover the higher convergence orders predicted by our theoretical numerical analysis.

The main message of the paper is that NeRF rendering should be treated with the same numerical care routinely expected in scientific computing. Our objective is not to claim a novel or unique numerical scheme for all NeRF-style renderers, but to show that many published formulas and corresponding implementations are mathematically inconsistent or suboptimal when judged through

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

root-mean-square convergence, and to document the smallest fixes that recover the highest possible convergence order.

Our contributions are as follows:

1. we prove an *a priori* error bound linking the NeRF rendering error with the error of the chosen VI quadrature rule;
2. we formalize two numerical issues in influential NeRF frameworks, explain theoretically why they reduce the root-mean-square convergence order, and introduce a minimal fix for each case study;
3. we validate the theoretical analysis with two complementary experimental settings: convergence tests on closed-form constant-color test cases and full NeRF training on standard benchmarks.

2 Background: volumetric rendering as a numerical problem

NeRFs are rooted in the classical theory of radiative transfer in participating media [6, 13]. Along a ray, light is attenuated by absorption and increased by emission. Let x denote the ray parameter on an interval $[x_n, x_f]$, let $\rho(x) \geq 0$ be the density, and let $\kappa(x)$ be the emitted color. The radiance $I(x)$ is modeled by the following first-order linear *final value problem*:

$$\begin{cases} \dfrac{dI}{dx} = \rho(x)I(x) - \rho(x)\kappa(x) & \forall\, x \in [x_n, x_f], \\ I(x_f) = I_f, \end{cases} \tag{1}$$

where I_f is the background radiance, that, in the rest of this work, we will consider equal to zero. Its solution at the near point x_n is the volumetric integral (VI):

$$I(x_n) = \int_{x_n}^{x_f} e^{-\int_{x_n}^{x} \rho(y)\, dy} \rho(x)\kappa(x)\, dx. \tag{2}$$

In a practical NeRF pipeline, the density field ρ and color field κ are represented by neural networks, typically MLPs. These learned fields are then queried along each camera ray and used in Eq. (2) to get the final rendered pixel color. However, Eq. (2) is computationally inconvenient because it contains a nested integral. In order to actually compute it, the ray domain is partitioned into K "cells"

$$[x_n, x_f] = [t_0, t_1] \cup [t_1, t_2] \cup \cdots \cup [t_{K-1}, t_K] \tag{3}$$

with $t_0 = x_n$ and $t_K = x_f$. The partition is assumed to be regular, i.e., $h := \max_k |t_{k-1} - t_k| \sim \frac{1}{K}$ for $K \to +\infty$. After exploiting the linearity of the integral domain, Eq. (2) can be equivalently expressed as:

$$I(x_n) = \sum_{k=1}^{K} \exp\left(-\sum_{j=1}^{k-1} \int_{t_{j-1}}^{t_j} \rho(s)ds \right) \left[\int_{t_{k-1}}^{t_k} e^{-\int_{t_{k-1}}^{t} \rho(s)ds} \kappa(t)\rho(t)dt \right] \tag{4}$$

To get from Eq. (4) to the usual NeRF rendering formula, the chromaticity has to be assumed constant over each cell:

$$\kappa(x) \approx \kappa_k \text{ for } x \in [t_{k-1}, t_k], \tag{5}$$

so that it can be pulled out of the integral and, by the fundamental theorem of calculus [1], it holds that:

$$\int_{t_{k-1}}^{t_k} e^{-\int_{t_{k-1}}^{t} \rho(s)ds} \kappa(t)\rho(t)dt = \left(1 - e^{-\int_{t_{k-1}}^{t_k} \rho(s)ds} \right) \kappa_k\,, \tag{6}$$

so that:

$$I := \sum_{k=1}^{K} \exp\left(-\sum_{j=1}^{k-1} \int_{t_{j-1}}^{t_j} \rho(s)\, ds \right) \left(1 - e^{-\int_{t_{k-1}}^{t_k} \rho(s)\, ds} \right) \kappa_k\,. \tag{7}$$

Finally, by choosing a numerical quadrature scheme to obtain approximations $\widehat{T}_k$ of the "optical depths"

$$T_k := \int_{t_{k-1}}^{t_k} \rho(s)\, ds \qquad \forall\, k = 1, \ldots, K\,, \tag{8}$$

the usual NeRF-style rendering formula is obtained:

$$\widehat{I} := \sum_{k=1}^{K} \exp\left(-\sum_{j=1}^{k-1} \widehat{T}_j\right) \left(1 - e^{-\widehat{T}_k}\right) \kappa_k \,. \tag{9}$$

The present paper does not modify the network architecture, positional encoding, training loss, or hierarchical resampling logic beyond the quadrature issues discussed later. Our focus is solely on the numerical integration layer that transforms pointwise field evaluations into rendered pixel values.

In practice, it is common [15] to use two sets of MLP predictions, usually referred to as "coarse" and "fine", together with a second rendering pass driven by importance resampling. The coarse pass provides a first estimate of the transmittance and weights along the ray, which are then used to define a CDF-like sampling distribution that allocates more samples to informative regions in the fine pass. Since our goal is to study the asymptotic behavior of the underlying quadrature rules, we defer this hierarchical machinery to the experimental section: it improves sample allocation, but it does not by itself resolve the consistency issues discussed in the following nor alter the convergence order induced by an incorrect local quadrature scheme.

In the rest of this work, we measure convergence in the *root-mean-square (RMS) sense* [8]. Let I be the target rendered value, and let $\widehat{I}$ be the corresponding discrete renderer on a partition with mesh size h.

Definition 1. *We say that $\widehat{I}$ has RMS order p if*

$$\mathbb{E}\left[|\widehat{I} - I|^2\right] = O(h^{2p}) \qquad \text{as } h \to 0. \tag{10}$$

We remark that the expectation is needed in the definition of Eq. (10) because the approximations $\widehat{T}_k$ could be stochastic variables.

PL-NeRF [24] revisits numerical integration in NeRF by replacing the standard piecewise-constant density assumption with a piecewise-linear model, hence modeling with greater accuracy the density field. In contrast, NeRFix retains the standard piecewise-constant approximation and focuses on the correctness and convergence of the quadrature rule itself. The two approaches are therefore orthogonal and potentially compatible: the NeRFix approach can also be used to verify the quadrature schemes of higher-order density models.

3 Methodology

In this section we present the two main theoretical tools we develop for the convergence analysis. The first is an *a priori* error bound that links NeRF rendering error with the error of the chosen VI quadrature rule, showing that a better quadrature rule results in lower rendering errors. The second is the analytical solution of the VI for a very simple scene with constant color, from now on referred to as the *closed-form constant-color test case*. We can use this setting to test the quadrature rule implementations, experimentally measure the rendering error at different sample counts, and verify that the convergence order matches the one prescribed by the *a priori* error bound.

In order to focus on the convergence study of the quadrature rules, throughout this work we assume that the approximation errors of the learned density and color fields are negligible, as is reasonable near the end of training, and we treat the standard interval-constant color simplification as not rate-limiting for the diagnostics considered here. This viewpoint isolates the rendering discretization from network approximation and optimization, and it makes clear that the errors studied here are numerical errors: they arise even when the underlying density and color fields are assumed known. We acknowledge that these errors could be partially absorbed by the training dynamics (i.e., the NeRF network still renders plausible images), but they remain present and measurable.

3.1 A priori error bound

In the next proposition we show that the RMS convergence order of the renderer is controlled by the RMS order of the chosen numerical integration scheme for the optical depths. This means that improving the quadrature rule should result in a lower rendering error.

We remark that Proposition 1 is intentionally generic: it does not prescribe a unique quadrature rule, but only states that the asymptotic behavior of the renderer is inherited from the asymptotic behavior of the chosen approximation of the optical depths.

Proposition 1. *It holds that*

$$\mathbb{E}\left[|\widehat{I} - I|^2\right] \leq c\mathbb{E}\left[\left|\sum_{k=1}^{K} |\widehat{T}_k - T_k|\right|^2\right] \tag{11}$$

where c is a positive constant.

Proof. Let's consider the following two recurrence relations that retrace the steps of the back-to-front compositing algorithm:

$$\begin{cases} \hat{V}_{K+1} = 0 \\ \hat{V}_k = \left(1 - e^{-\widehat{T}_k}\right)\kappa_k + e^{-\widehat{T}_k}\hat{V}_{k+1} \qquad \forall\, k = K, \ldots, 1 \end{cases} \tag{12}$$

$$\begin{cases} V_{K+1} = 0 \\ V_k = \left(1 - e^{-T_k}\right)\kappa_k + e^{-T_k}V_{k+1} \qquad \forall\, k = K, \ldots, 1\,. \end{cases} \tag{13}$$

It is easy to verify that $V_1 = I$ and $\hat{V}_1 = \widehat{I}$, so that $|\widehat{I} - I| = |V_1 - \hat{V}_1|$.

The k-th difference can be rewritten recursively as:

$$|V_k - \hat{V}_k| = |\left(1 - e^{-T_k}\right)\kappa_k + e^{-T_k}V_{k+1} - \left(1 - e^{-\widehat{T}_k}\right)\kappa_k - e^{-\widehat{T}_k}\hat{V}_{k+1}| \tag{14}$$

$$= |\left(e^{-\widehat{T}_k} - e^{-T_k}\right)\kappa_k + e^{-T_k}V_{k+1} - e^{-\widehat{T}_k}\hat{V}_{k+1}| \tag{15}$$

$$= |\left(e^{-\widehat{T}_k} - e^{-T_k}\right)\kappa_k + e^{-T_k}V_{k+1} - e^{-\widehat{T}_k}\hat{V}_{k+1} + e^{-T_k}\hat{V}_{k+1} - e^{-T_k}\hat{V}_{k+1}| \tag{16}$$

$$= |e^{-T_k}\left(V_{k+1} - \hat{V}_{k+1}\right) + \left(e^{-\widehat{T}_k} - e^{-T_k}\right)\left(\kappa_k - \hat{V}_{k+1}\right)| \tag{17}$$

$$\leq e^{-T_k}|V_{k+1} - \hat{V}_{k+1}| + |e^{-\widehat{T}_k} - e^{-T_k}||\kappa_k - \hat{V}_{k+1}|\,, \tag{18}$$

where we used the triangle inequality in the last step.

Let B be a constant such that $|\kappa_k| \leq B\,\forall\, k$. Then it holds that $|\hat{V}_k| \leq B\,\forall\, k$. This can be proved by induction: the base step is trivial ($\hat{V}_{K+1} = 0$) and the inductive step follows from the definition of $\hat{V}_k$ being a convex combination whose endpoints, κ_k and $\hat{V}_{k+1}$, are by definition smaller than B in absolute value. Hence $|\kappa_k - \hat{V}_{k+1}| \leq |\kappa_k| + |\hat{V}_{k+1}| \leq 2B$.

Exploiting the fact that T_k is positive (as $\rho \geq 0$ in Eq. (1)) so that $e^{-T_k} \leq 1$, we get:

$$|V_k - \hat{V}_k| \leq |V_{k+1} - \hat{V}_{k+1}| + 2B|e^{-\widehat{T}_k} - e^{-T_k}|\,. \tag{19}$$

Remembering that $|V_{K+1} - \hat{V}_{K+1}| = 0$, it holds that:

$$|\widehat{I} - I| = |V_1 - \hat{V}_1| \leq 2B\sum_{k=1}^{K}|e^{-\widehat{T}_k} - e^{-T_k}|\,. \tag{20}$$

The proof is completed by remembering that $|(e^{-x})'| \leq 1\ [0, +\infty)$, so that, by the mean value theorem [1], it holds that $|e^{-\widehat{T}_k} - e^{-T_k}| \leq |\widehat{T}_k - T_k|$.

□

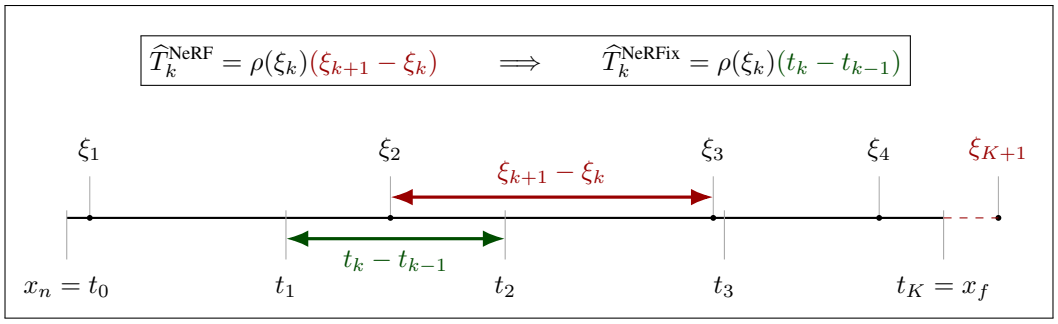

Figure 1: Original NeRF: numerical inconsistency and minimum fix. The top row shows the optical-depth approximation used in the original formulation and its corrected version. The problematic term is the interval length: the original rule uses the distance between adjacent random samples, $\xi_{k+1} - \xi_k$, whereas the fix uses the width of the physical cell, $t_k - t_{k-1}$. Moreover, the term ξ_{K+1} is not defined. The bottom row illustrates the geometry on the full domain $[x_n, x_f]$, partitioned into equispaced cells with one random sample per cell.

3.2 Closed-form constant-color test cases

To validate the theory, we experimentally measure convergence orders using a sanity-check scenario in which (i) the color is constant along the whole ray, and (ii) the function ρ admits a primitive P. In this case, the solution of Eq. (2) has the following closed form:

$$I(x_n) = \kappa(x) \int_{x_n}^{x_f} e^{-\int_{x_n}^{x} \rho(y)\, dy} \rho(x)\, dx = \left(1 - e^{-(\mathrm{P}(x_f) - \mathrm{P}(x_n))}\right) \kappa. \tag{21}$$

For example, if $\kappa(x) = \kappa_0$ for all $x \in [x_n, x_f]$ and $\rho(x) = \rho_0$ for all $x \in [x_n, x_f]$, then:

$$I = \kappa_0 \left(1 - e^{-\rho_0 (x_f - x_n)}\right). \tag{22}$$

This explicit formula gives an exact reference value against which we can compare different rendering schemes. In practice, we render the same ray with increasing numbers of samples, compute the error with respect to the closed-form solution, and fit a straight line in a log–log plot of the rendering error $\mathbb{E}\left[|\widehat{I} - I|^2\right]$ versus the number of samples K. The fitted slope provides the empirical RMS convergence order, that ought to match the order prescribed by Proposition 1, as long as the numerical scheme is correctly implemented.

4 NeRF mistakes and their fixes

In the next two sections we study the quadrature schemes used in the literature through the lens of Proposition 1, reporting the current convergence orders and computationally-free adjustments (i.e., their usage incurs in no additional computational costs) to achieve higher convergence orders.

4.1 Case study I: the original NeRF

The numerical integration scheme proposed in the original NeRF paper [15] is, read as written, not well-defined, and, in the closed-form constant-color test case, its public reference implementation is not convergent to Eq. (22).

In the original paper, the partition of Eq. (3) is equispaced:

$$[t_{k-1}, t_k] := \left[x_n + \frac{k-1}{K}(x_f - x_n), x_n + \frac{k}{K}(x_f - x_n)\right] \qquad \forall\, k = 1, \ldots, K\,. \tag{23}$$

Then, one point is sampled per cell:

$$\xi_k \sim \text{Uniform}[t_{k-1}, t_k] \qquad \forall\, k = 1, \ldots, K\,, \tag{24}$$

and the following approximation scheme is chosen:

$$\widehat{T}_k^{\text{NeRF}} = \rho(\xi_k)\,(\xi_{k+1} - \xi_k) \qquad \forall\, k = 1, \ldots, K\,; \tag{25}$$

the resulting rendered pixel intensity is therefore:

$$\widehat{I}^{\text{NeRF}} = \sum_{k=1}^{K} \exp\left(-\sum_{j=1}^{k-1} \rho(\xi_j)\,(\xi_{j+1} - \xi_j)\right)\left(1 - e^{-\rho(\xi_k)(\xi_{k+1}-\xi_k)}\right)\kappa_k\,. \tag{26}$$

However, only $\xi_1, \ldots, \xi_K$ are defined (i.e., one per cell), so the last quantity $\widehat{T}_K^{\text{NeRF}}$, that depends on ξ_{K+1}, does not exist. The published formula is therefore not well-defined. The reference implementation tries to resolve this ambiguity by arbitrarily setting $\xi_{K+1} := 10^{10}$. It can be shown that this scheme is not convergent in the closed-form constant-color test case. It is easy to check by telescoping and basic algebra that Eq. (26), evaluated for constant density and constant color, results in $\kappa_0(1 - e^{-\rho_0(\xi_{K+1}-\xi_1)})$, which is not convergent to the true analytical solution in Eq. (22), as $\xi_{K+1} \nrightarrow x_f$.

The root cause of this mistake is probably a mismatch between the given interpretation of Eq. (25) and what it is actually computing. There are two competing interpretations. On the one hand, the formula can be read as a Monte Carlo estimator on a fixed deterministic partition. On the other hand, if one interprets the sampled points themselves as defining a random partition, then the same expression resembles a classical deterministic quadrature rule applied on a stochastic mesh.

Monte Carlo quadrature interpretation. In the original paper the scheme was intended as a Monte Carlo method with one sample per cell, i.e., a stochastic method applied to a deterministic partition. In this interpretation, the key mistake is that the integration widths are taken to be distances between adjacent random samples rather than widths of the physical intervals from which those samples were drawn. The minimum fix is to keep the same random point in each physical interval but to replace the sample-to-sample widths by the true cell widths. Importantly, this correction does not introduce any additional computational overhead. The same MLP evaluations are performed at the same sampled locations, and the only change is the scalar interval length used to weight each density evaluation. In compact form,

$$\widehat{T}_k^{\text{NeRFix}} = \rho(\xi_k)\,(t_k - t_{k-1}) \qquad \forall\, k = 1, \ldots, K\,. \tag{27}$$

A graphical depiction of the problem and the fix is shown in Fig. 1.

Left-rectangle rule interpretation. Eq. (25) has also another interpretation: a deterministic method applied to a stochastic partition. Indeed Eq. (25) can be interpreted, after skipping the last inconsistent term ξ_{K+1}, as the well-known [8] (deterministic) left-rectangle quadrature rule applied onto the (stochastic) partition $[\xi_1, \xi_2] \cup \ldots \cup [\xi_{K-1}, \xi_K]$.

More generally, regardless of ξ_{K+1}, Proposition 1 prescribes that the original method of Eq. (26) has convergence order of 1 as its numerical scheme, the left-rectangle rule, has convergence order of 1 [8]. Instead, the improved version in Eq. (27) has an RMS order of $\frac{3}{2}$, as it is a proper Monte-Carlo integration scheme [8].

In the case where a deterministic output is preferable, for example after training, when the learned MLPs are frozen and one wishes to render test views without stochastic variability, the same correction naturally suggests a deterministic counterpart. Namely, one can place the evaluation point at the midpoint of each physical cell and keep the correct cell width in Eq. (27). This yields the classical midpoint rule on the original partition, removing sampling noise altogether and recovering second-order convergence [8]. Thus, the proposed fix is compatible both with the stochastic setting used during training and with a deterministic high-accuracy renderer at test time.

4.2 Case study II: NeRFStudio

NeRFStudio (NS) is a particularly relevant case study because it has become one of the most widely adopted open-source frameworks for developing, benchmarking, and extending NeRF-style methods [23]. Its popularity comes from offering a modular and well-engineered

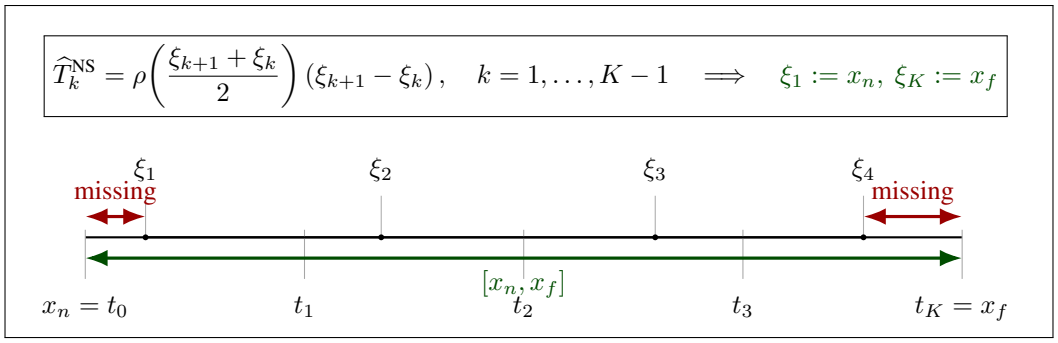

Figure 2: NeRFStudio: numerical inconsistency and minimum fix. The top row shows the midpoint-based optical-depth approximation together with the endpoint correction used in the fixed version. The local midpoint rule is unchanged; the fix consists in forcing the random chain to span the full domain by setting $\xi_1 = x_n$ and $\xi_K = x_f$. The bottom row shows the geometry: the original construction covers only the truncated interval $[\xi_1, \xi_K]$, leaving the two endpoint pieces uncovered, whereas the fixed version restores the full domain $[x_n, x_f]$.

codebase that makes it easy to compare rendering variants, plug in new models, and reproduce standard pipelines across datasets. For this reason, numerical choices implemented in NS are likely to propagate broadly through subsequent work. Analyzing its rendering rule is therefore important not only for the correctness of one specific implementation, but also because NS effectively serves as a backend for a substantial portion of the current NeRF ecosystem. Here we analyze the `generate_ray_samples` method of `SpacedSampler` in `nerfstudio/model_components/ray_samplers.py`, as used by its `vanilla-nerf` path. Samplers that inherit from this class are: `UniformSampler`, `LinearDisparitySampler`, `SqrtSampler`, `LogSampler`, and `UniformLinDispPiecewiseSampler`. The NeRFStudio methods affected at the code level, at least in the coarse network (without claiming or excluding a performance gap in end-to-end training), include `vanilla-nerf`, `dnerf`, `mipnerf`, `tensorf`, `nerfacto`, `nerfacto-big`, `nerfacto-huge`, `phototourism`, `depth-nerfacto`, `semantic-nerfw`, `generfacto`, `neus`, and `neus-facto`.

NS proceeds similarly to the original NeRF paper, considering an equispaced partition of $[x_n, x_f]$ and sampling uniformly one point per interval:

$$\xi_k \sim \text{Uniform}\left[x_n + \frac{k-1}{K}(x_f - x_n), x_n + \frac{k}{K}(x_f - x_n)\right] \qquad \forall\, k = 1, \dots, K\,. \tag{28}$$

Then, NS uses the following approximation scheme:

$$\widehat{T}_k^{\text{NS}} = \rho\left(\frac{\xi_{k+1}+\xi_k}{2}\right)(\xi_{k+1}-\xi_k) \qquad \forall\, k = 1, \dots, K-1\,, \tag{29}$$

where the number of samples is effectively reduced by one, solving the well-posedness mistakes presented in the previous section. Therefore, the overall rendering scheme is:

$$\widehat{I}^{\text{NS}} = \sum_{k=1}^{K-1} \exp\left(-\sum_{j=1}^{k-1} \rho\left(\frac{\xi_{j+1}+\xi_j}{2}\right)(\xi_{j+1}-\xi_j)\right)\left(1 - e^{-\rho\left(\frac{\xi_{k+1}+\xi_k}{2}\right)(\xi_{k+1}-\xi_k)}\right)\kappa_k\,. \tag{30}$$

This formulation has a similar interpretation to the left-rectangle rule of the previous section, but it instead uses the deterministic mid-point rule applied to the stochastic partition induced by the ξ_k's. It is well-known that the mid-point rule yields a second-order accurate scheme, hence it is strictly better than the left-rectangle rule. However, this extra performance is not fully exploited because the proposed method has again a numerical mistake, as the integrated domain is effectively $[\xi_1, \xi_K]$ instead of $[x_n, x_f]$, i.e., the numerical scheme is missing $[x_n, \xi_1]$ and $[\xi_K, x_f]$, limiting again the convergence order. Because of the assumption of the partition being regular – cf. Eq. (3)– the lengths of the two missing intervals converge to zero as h, so the overall convergence order of the scheme is 1. We also remark that in the non-asymptotic setting, scene content might be present in the neglected

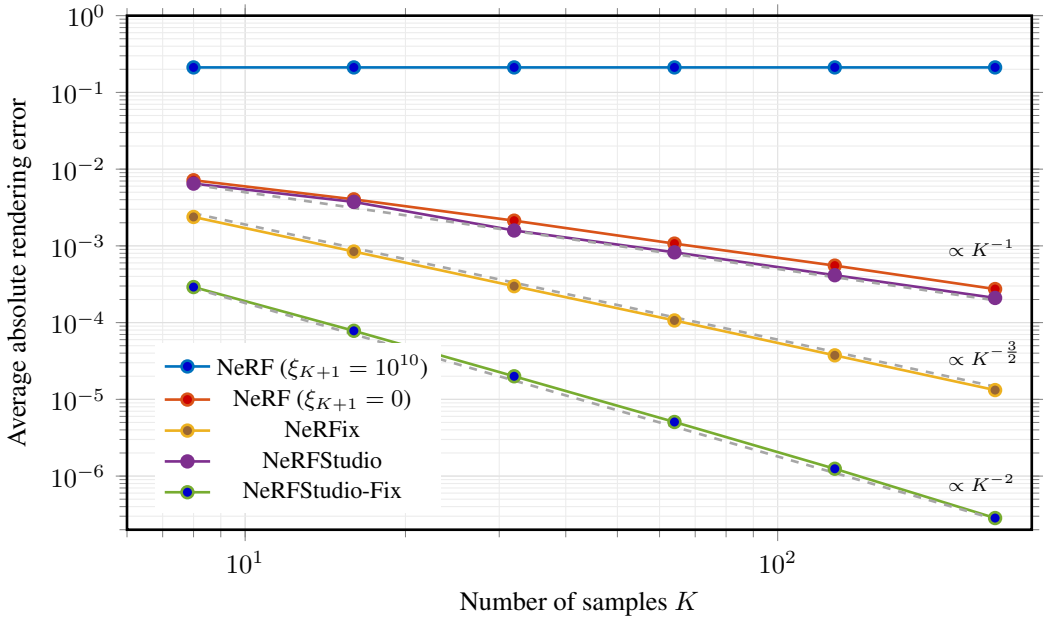

Figure 3: Synthetic convergence on the test problem, shown as a log–log plot of average absolute error versus the number of samples K. The original NeRF without fixing the ξ_{K+1} mistake does not converge. The NeRF method obtained by setting it to zero, and the NeRFStudio discretizations exhibit approximately first-order behavior, with fitted slopes -0.947 and -0.982, respectively. The proposed fixes recover the higher orders predicted by the theory, with fitted slopes -1.498 for NeRFix and -1.987 for NeRFStudio-Fix. Dashed lines indicate the reference rates K^{-1}, $K^{-\frac{3}{2}}$, and K^{-2}.

intervals, leading to rendering errors. The minimum fix is to forcefully set $\xi_1 = x_n$ and $\xi_K = x_f$, keeping the number of MLP samples constant and also recovering the lost second-order convergence. A graphical depiction of the problem and the fix is shown in Fig. 2.

5 Experiments

5.1 Exact rendering of a constant-color scene

We first validate the theory on synthetic one-dimensional test problems where the reference solution is known exactly. Our main diagnostic is the constant-color setting described in Eq. (22), because it isolates the discretization error and allows a direct measurement of the RMS convergence order. In particular, we arbitrarily choose the following functional form for $\rho(x) = \cos\left(\frac{\pi}{2}x\right)$.

For each method, we render the same ray with increasing numbers of samples, compute the error with respect to the closed-form solution, and fit a straight line in a log–log plot of the rendering error versus the sample number K, whose slope represents the convergence order. The results are shown in Fig. 3, and they confirm the theoretical analysis. The original NeRF (even when fixing $\xi_{K+1} = 0$) and NeRFStudio schemes do not exceed first-order RMS behavior. In contrast, the minimum fix for the original NeRF case recovers the expected 1.5 higher-order behavior, while the corresponding fix for the NeRFStudio case restores the second-order convergence. Beyond the asymptotic slopes, the fixed variants also yield smaller errors at practical sample counts. This matters because NeRF training always operates in a finite-sample regime rather than in an asymptotic limit.

5.2 Full NeRF training on standard benchmarks

We now turn our attention to assessing the impact of the proposed fixes in practical settings. Table 1 compares the original NeRF renderer with our minimal fix in full end-to-end training on well-known datasets: Blender [15], DeepVoxels [22], and LLFF [14]. For fairness, we re-run all methods under exactly matched conditions: same train/test splits (the same ones used in the original NeRF

paper), same optimizer hyperparameters (Adam with $\beta_1 = 0.9, \beta_2 = 0.999$), same learning rate scheduler (exponential decay from 5×10^{-4} to 5×10^{-5}), same coarse/fine sampling budgets (64 and 128, respectively), and same network architecture (the standard 8-layer 256-wide MLP with a skip connection, a density head, and a view-dependent RGB head). The only change is the numerical quadrature scheme. For each scene and method, we perform 10 runs; we first calculate each scene's mean and standard deviation across runs, then average those statistics across scenes in each dataset. The fix improves both PSNR and SSIM on all three benchmarks, yielding consistent gains without introducing any additional computational cost (on the order of 20 hours on a single V100 GPU for each scene run).

Our main observation is that the minimum fixes yield small but consistent gains, improving test-set PSNR by approximately 0.1 dB and a positive fractional amount of SSIM, without any computational overhead. This confirms that the issue is not purely academic: even though a better quadrature scheme is local, it propagates through training and produces a better learned radiance field. Moreover, we stress the fact that these fixes do not incur in any computational overhead and are theoretically grounded.

6 Conclusion

We revisited the numerical core of NeRF rendering and showed that small inconsistencies in the quadrature scheme can have measurable theoretical and practical consequences. We analyzed two influential case studies. In the original NeRF paper, the estimator is not well-posed as written, and the reference implementation replaces physical cell widths with distances between adjacent random samples. In NeRFStudio, the midpoint-based construction is applied on a random partition that does not fully cover the integration domain because the two endpoint intervals are omitted.

For both cases, we introduced a minimum fix: the smallest possible modification that removes the specific inconsistency while preserving the original structure of the renderer. In the original NeRF case, the fix is to weight each sampled density by the width of its physical cell rather than by the distance to the next random sample. In the NeRFStudio case, the fix is to restore the missing endpoint intervals so that the midpoint construction again spans the full domain.

Our theoretical analysis and synthetic experiments show that these details directly affect RMS convergence. The corrected schemes recover the higher convergence orders predicted by the quadrature analysis, whereas the original formulations are limited to lower-order behavior or, in the inconsistent NeRF case, may even fail to converge. Practical experiments on Blender, LLFF, and DeepVoxels further show that these fixes are not only theoretically cleaner, but also beneficial in training, yielding small but consistent improvements in image quality. A gain of about 0.1 dB is modest, but it is still notable because it is obtained without adding parameters, samples, or computational cost.

Limitations. Our analysis isolates the numerical quadrature layer of NeRF rendering and does not model other important sources of error such as imperfect learned density and color fields, optimization effects, or architecture-dependent training dynamics. The experimental evaluation is limited to a small set of standard benchmarks, so the reported gains should be interpreted as evidence that the fixes matter in practice, rather than as a claim of uniformly large improvements across all NeRF variants. More generally, although the two case studies considered here are influential and representative, our results do not imply that every NeRF-style renderer in the literature exhibits exactly the same inconsistencies.

Table 1: Full NeRF training results on test sets, comparing the original [15] and the fixed version.

	PSNR (dB) ↑			SSIM ↑		
Dataset	Original	Fixed	Δ	Original	Fixed	Δ
Blender	31.93 ± 0.05	$\mathbf{32.08 \pm 0.07}$	+0.15	0.9506 ± 0.0004	$\mathbf{0.9516 \pm 0.0004}$	+0.0010
DeepVoxels	41.10 ± 0.08	$\mathbf{41.28 \pm 0.09}$	+0.19	0.9906 ± 0.0001	$\mathbf{0.9909 \pm 0.0001}$	+0.0003
LLFF	25.44 ± 0.06	$\mathbf{25.45 \pm 0.06}$	+0.02	0.7950 ± 0.0011	$\mathbf{0.7952 \pm 0.0011}$	+0.0002
Average	32.82 ± 0.06	$\mathbf{32.94 \pm 0.07}$	+0.12	0.9121 ± 0.0005	$\mathbf{0.9126 \pm 0.0006}$	+0.0005

Within this scope, the main conclusion is simple: NeRF rendering should be viewed not only as a differentiable layer, but also as a numerical method. Its asymptotic behavior depends on the correctness of the underlying quadrature scheme, and even very small same-complexity implementation choices can determine whether the renderer is mathematically consistent, convergent, and ultimately slightly better in practice.

Funding and Competing Interests

The authors received no third-party funding or support for this work and declare no competing financial interests.

References

[1] T. M. Apostol. *Calculus / Vol. 1, One-variable calculus, with an introduction to linear algebra.* John Wiley, New York, 2nd ed. edition, 1967. ISBN 9780471000068.

[2] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021.

[3] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022.

[4] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *arXiv preprint arXiv:2304.06706*, 2023.

[5] W. Bian, Z. Wang, K. Li, and J.-W. Bian. Nope-nerf: Optimising neural radiance field with no pose prior. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4160–4169, 2023. doi: 10.1109/CVPR52729.2023.00405.

[6] S. Chandrasekhar. *Radiative transfer*. Courier Corporation, 2013.

[7] X. Chen, Q. Zhang, X. Li, Y. Chen, Y. Feng, X. Wang, and J. Wang. Hallucinated neural radiance fields in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12943–12952, 2022.

[8] P. J. Davis and P. Rabinowitz. *Methods of numerical integration*. Courier Corporation, 2007.

[9] B. K. Isaac-Medina, C. G. Willcocks, and T. P. Breckon. Exact-nerf: An exploration of a precise volumetric parameterization for neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 66–75, 2023.

[10] Y. Jeong, S. Ahn, C. Choy, A. Anandkumar, M. Cho, and J. Park. Self-calibrating neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5846–5854, 2021.

[11] C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey. Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5741–5751, 2021.

[12] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7210–7219, 2021.

[13] N. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995. doi: 10.1109/2945.468400.

[14] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (ToG)*, 38(4):1–14, 2019.

[15] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[16] B. Mildenhall, P. Hedman, R. Martin-Brualla, P. P. Srinivasan, and J. T. Barron. Nerf in the dark: High dynamic range view synthesis from noisy raw images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16190–16199, 2022.

[17] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.

[18] M. Niemeyer, J. T. Barron, B. Mildenhall, M. S. Sajjadi, A. Geiger, and N. Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022.

[19] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla. Nerfies: Deformable neural radiance fields. *ICCV*, 2021.

[20] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[21] C. Reiser, S. Peng, Y. Liao, and A. Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14335–14345, 2021.

[22] V. Sitzmann, J. Thies, F. Heide, M. Nießner, G. Wetzstein, and M. Zollhöfer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proc. CVPR*, 2019.

[23] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 conference proceedings*, pages 1–12, 2023.

[24] M. A. Uy, K. Nakayama, G. Yang, R. Thomas, L. Guibas, and K. Li. NeRF revisited: Fixing quadrature instability in volume rendering. In *Advances in Neural Information Processing Systems*, volume 36, pages 26159–26189, 2023. doi: 10.52202/075280-1139.

[25] Z. Wang, S. Wu, W. Xie, M. Chen, and V. A. Prisacariu. NeRF$--$: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the main claims of the paper: identifying two quadrature inconsistencies in influential NeRF renderers, proving an a-priori rendering error bound, proposing minimum fixes, and showing both convergence improvements and modest practical gains. These claims are reflected in Sections 1, 3, 4, 5, and 6.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper explicitly discusses its limitations in the conclusion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The paper provides a complete proof of Proposition 1 in the main text.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The experimental results are based on open-source codebases and the modification to those codebases are described in the main text.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code is available at `https://github.com/IPLLab-PoliTO/nerfix`.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper uses the standard setting introduced by the original NeRF paper by Mildenhall et al. and re-reports them in the Experiments section.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Section 5.2 and Table 1 report means and run-to-run standard deviations from 10 runs per scene, then average these statistics across scenes for each dataset.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Each scene requires about 20 hours on a single V100 GPU, as described in the Experiments section.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work is a methodological study of numerical quadrature in existing NeRF renderers and does not involve interventions that conflict with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The current paper does not include a dedicated discussion of broader societal impacts, because it is primarily a numerical analysis of rendering accuracy, hence it has no societal impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper is mostly theoretical. Experiments use publicly available datasets and implementations which are cited in the Experimental section.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human-subject research or crowdsourcing, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not part of the core method, experiments, or theoretical contribution of this work.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.