SABRe-NeRF: Specular Attention Based Rendering for Neural Radiance Fields

Silas A. Mohr
Princeton University
smohr@princeton.edu

Abstract

Neural radiance fields (NeRF), and its descendants, has been the leading method of novel view synthesis since its introduction in 2020. It used a 3D continuous volumetric representation by way of hyper over-fitting a single multilayer perceptron to a single scene. However, it struggles with reflections because of the traditional volumetric rendering pipeline that it used. One paper, Ref-NeRF aimed to remedy this by prediction normals and explicity calculating reflected viewing directions. Another paper, ABLE-NeRF used a transformer based approach to learn the rendering equation. This paper introduces specular attention based rendering for neural radiance fields (SABRe-NeRF), which combines the two techniques. I found that it outperforms the baseline Ref-NeRF slightly and visually appears more consistent with the ground truth. More work tuning the hyperparameters is necessary however, and future work could find increases in performance by exploring that process.

1 Introduction

The field of neural rendering has undergone massive innovation in the last few years, especially after the release of the paper that introduced the concpet of neural radiance fields for novel view synthesis, also known NeRF, by Mildenhall et al. (2020). NeRF used a single, fully-connected and non-convolutional deep network to represent a 3D scene. The network took a single continuous 5D input coordinate, comprised of a 3D spatial coordinate and a 2D viewing direction, as input, which allowed the network to be queried with novel camera positions and angles, thus providing a method for novel view synthesis. Many other papers have built off of this architecture, like Mip-NeRF (Barron et al., 2021), which improved upon the ray sampling technique, and Ref-NeRF (Verbin et al., 2021), which strengthened the ability of NeRF to handle reflections.

Some papers have taken the improvements in a different direction and started to introduce methods from other fields of machine learning, like using transformers that were originally used within natural language processing (Wang et al., 2021; Tang et al., 2023).

This paper aims to combine the optimizations from both paths to create a model that produces accurate reflections for both smooth and complex objects. I propose using a set of learned embeddings to learn scene information along with the normals and using a transformer to learn the attention needed to use information from both the learned embeddings and normals similar to Tang et al. (2023).

2 Related Work

2.1 Physics Based NeRF Models

As discussed in Section 1, this paper builds heavily off of the advancements in neural rendering based off the model, NeRF, introduced by (Mildenhall et al., 2020). NeRF is based around a single multilayer perceptron (MLP) and uses a three step process to render novel views using what they

Preprint. Under review.

called neural radiance fields. The first step is marching rays from the camera through the scene and sampling along each ray to generate a set of 3D points. Next, these points are used as the 3D coordinates and the direction of the ray the point is from as the 2D viewing direction as input to the MLP as mentioned in Section 1. The MLP then outputs a set of colors and densities for these sampled points. Finally, the model uses traditional volume rendering techniques to create a 2D image from these sampled colors and densities.

However, while NeRF was crucial and substantial leap forward in neural rendering, it struggled when training or test images were at different resolutions, so Barron et al. (2021) published Mip-NeRF, a version of NeRF that used conical frustums to sample along the camera rays, rather than single points as shown in Fig 1.

Ref-NeRF is a model built off of Mip-NeRF that offered a method of enhancing reflections with three key additions to the baseline architecture (Verbin et al., 2021). The first of these was splitting the outgoing radiance at each point into incoming radiance, diffuse color, material roughness, and specular tint. These are all used explicity within Ref-NeRF's model as shown in 3a. predicting normals within the scene representation and reflecting the input viewing directions using said normals to parameterize the outgoing radiance rather than simply using the viewing directions. The final addition is using integrated directional encoding (IDE) to better Since Ref-NeRF relies on the predicted normals to create an accurately reflected viewing direction, as it uses the explicit reflection equation, the quality of said reflection is based on the quality of the predicted normals. Therefore, if the normals are not accurate in relation to the ground truth of the scene, the render quality will suffer with regards to ground truth faithfulness. This is apparent in scenes like the canonical Blender ship, which includes complex waves, as seen in Fig 2.

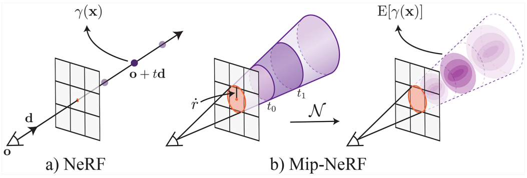


Figure 1: Visualization of Mip-NeRF's conical frustum sampling and integrated positional encoding $E[\gamma(x)]$ from (Barron et al., 2021).

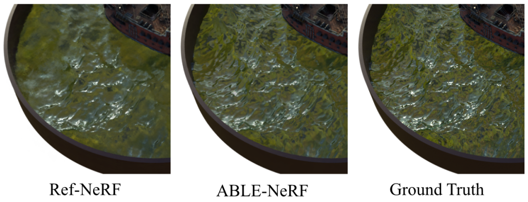


Figure 2: Renders of waves in the Blender ship scene. The Ref-NeRF render is blurred, especially in the top-left of the image. Figure is from (Tang et al., 2023)

2.2 Transformer Based Novel View Synthesis

The transformer, presented in the influential 2017 paper "Attention Is All You Need", offered a novel architecture based on attention mechanisms, replacing previous methods that used recurrent or convolutional neural networks within the field of natural language processing (NLP) (Vaswani et al., 2017). Transformers reduced complexity, training time, and improved the performance of language models, initially on language-to-language translation tasks.

By leveraging self and cross-attention, transformers excelled at capturing long-range dependencies between words and understanding the semantic relationships within a given text. This capability proved invaluable in improving the quality of language models, enabling them to generate context-aware representations of text and perform complex NLP tasks more accurately. Because of these strengths, transformers have found use cases within many other fields, including neural rendering, inspiring many papers to include them in their proposed models. One large language model that came out of this is BERT, which uses a class token for each sentence it takes in as input, which is a technique used by ABLE-NeRF and this paper (Devlin et al., 2018; Tang et al., 2023).

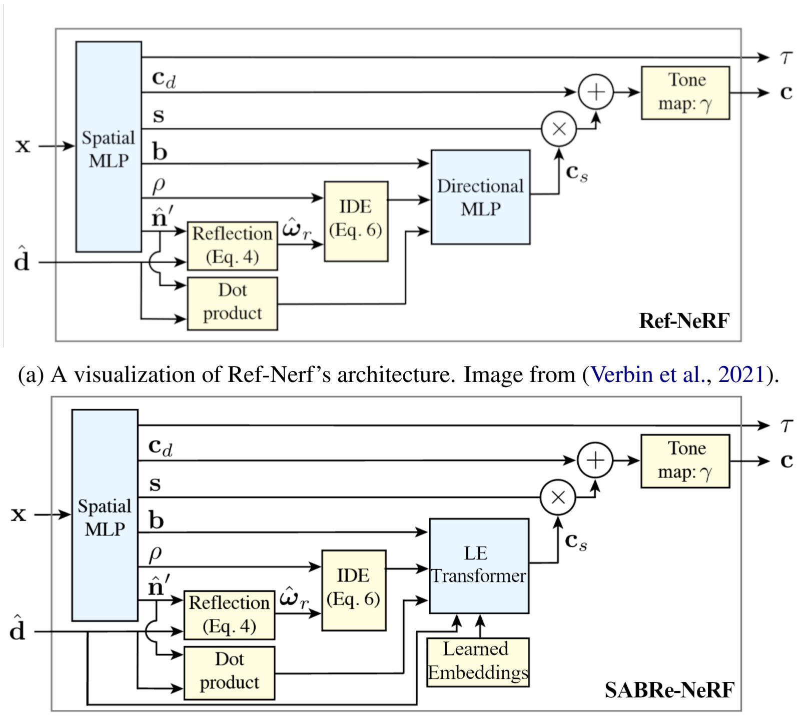
IBRNet is one such paper that introduced transformers into their rendering pipeline for novel view synthesis (Wang et al., 2021). IBRNet consists of a single MLP, similar to the baseline NeRF model, which then passes its output into a ray transformer to generate a density σ_i per ray. However, the model differs from NeRF in that it is meant to be a generalized model which can then be fine-tuned on specific scenes.

Another paper that used transformers for novel view synthesis is ABLE-NeRF (Tang et al., 2023). ABLE-NeRF's main point was to replace the explicitly physics based rendering equation with transformers to produce an implicitly physics based rendering pipeline. The concept of learnable embeddings to learn scene information that was not included within the geometry of the scene, like how light is passing through the air, i.e. light probes, was also introduced by the same paper.

The implementation used in this paper was written using Flax (Heek et al., 2023), a library written to facilitate neural network development based on JAX (Bradbury et al., 2018). This allowed for efficient distributed computation while training on multiple CPUs and GPUs.

3 Theory

The architecture proposed by this paper is substantially based on the model proposed by Verbin et al. (2021). Specifically, this paper only shifts from using a directional MLP that outputs a specular color and instead implements a learnable embeddings (LE) transformer that is based off of the LE Transformer from Tang et al. (2023) as seen in 3b. Since the transformer based architecture in ABLE-NeRF was better able to represent complex shapes than Ref-NeRF, the LE transformer, which outputs the specular color for ABLE-NeRF was chosen as the model to use in this paper. It's learned embeddings can offer an alternative when the normals of a shape are too complex for the model to accurately represent, thus allowing for better reflections for all kinds of objects.



(a) A visualization of Ref-Nerf's architecture. Image from (Verbin et al., 2021).

(b) A visualization of SABRe-NeRF's architecture. The

Figure 3: A comparison of SABRe-NeRF and Ref-Nerf's architectures.

3.1 Learnable Embeddings Transformer

The LE Transformer takes in the bottleneck output (b) directly from the spatial MLP, the output from IDE of the reflected viewing directions (ω_r) and roughness (ρ), the initial viewing direction ($\hat{\mathrm{d}}$), and the learned embeddings. As illustrated by Fig 4, the internal architecture consists of a first cross attention layer that cross attends between the learned embeddings and a vector of the concatenated IDE and bottleneck outputs. This is then passed through a self attention layer and through a final cross attention layer that takes in an embedded version of the original viewing direction as a token. The embedding is a simple fourier space embedding. This token is similar to the class token from

BERT (Devlin et al., 2018) and allows the transformer to better attend to parts of the scene depending on the viewing direction.The final cross attention layer then outputs a value for the specular color (c_s) which is used in the final calculations of the color.

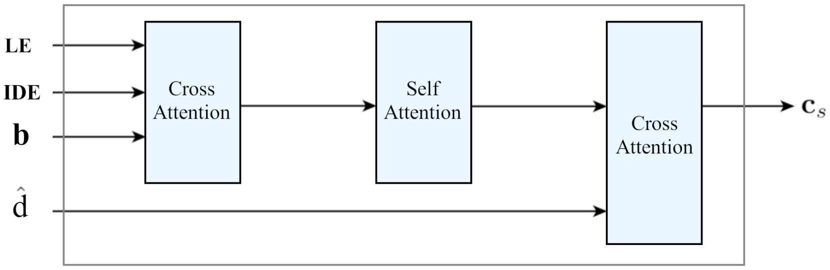


Figure 4: The internal LE Transformer architecture

4 Analysis

All training and experiments were performed using two CPU cores with 16GB of RAM and two Nvidia A100 GPUs with 40 GB of memory. To maintain manageable memory usage, I found that for training the batch size, i.e. the number of rays per training step, the value should be set to 4096, 2^{12}, as opposed to the 16384, 2^{14}, that is the default value. However, when training the baseline Ref-NeRF model, I found that a batch size of 8192, 2^{13}, was also possible. To fully match the results from the Ref-NeRF paper with a non-default batch size, the total number of learning iterations should be increased and the learning rate should be decreased by a factor of how much the batch size is changed. Therefore, with a batch size of 4096, the number of iterations should become $250\text{k} \cdot 4 = 1\text{m}$, and the learning rate should be $2 \cdot 10^{-4} \div 4 = 5 \cdot 10^{-5}$. However, training 10k iterations takes around 50 minutes, so training with 1m steps would take around four days, which was too long for this project. Thus, to compensate for the reduced batch size, the total number of iterations was increased from 250000 to 35000 and learning rate was lowered to $1.7 \cdot 10^{-4}$ from $2 \cdot 10^{-4}$, which took around 30 hours to train.

For the LE transformer, I used 100 light probes, each of which is a vector of length 128, so as to match the width of the bottleneck vector.

5 Results

5.1 Quantitative

From the metrics in Table 1, we can see that the paper outperforms Ref-NeRF (at least the baseline that I was able to reproduce), but struggles to match the quality of normals. I did not have the time to train the baseline model using the ball scene.

5.2 Qualitative

When looking at rendered images in Fig 5, the strengths of my proposed model are evident. The reflection of the toast on the top of the toaster is crisper and reads more closely to what we expect the reflection to look like, especially with regards to the difference in color of the crust. Another improvement is with the strength of the specular highlights. While the shape of the highlights is not perfect in either the Ref-NeRF image, Fig 5b, or this papers image, Fig 5c, the intensity of the highlight is matched much more closely in my models image. The reflections on the front of the toaster are also smoother and less noisy, closer to how they look on the ground truth image, Fig 5a.

6 Discussion and Conclusions

This paper did not fully explore the hyperparameters chosen for the model, so further experiments to tweak them are necessary to find the best performance. The timeline of this project also did not allow

Model	*toaster*	*ball*
Ref-NeRF (from the paper)	25.70	47.46
Ref-NeRF (mine)	23.35	-
SABRe-NeRF	24.16	35.53

(a) Per scene test set PSNRs↑.

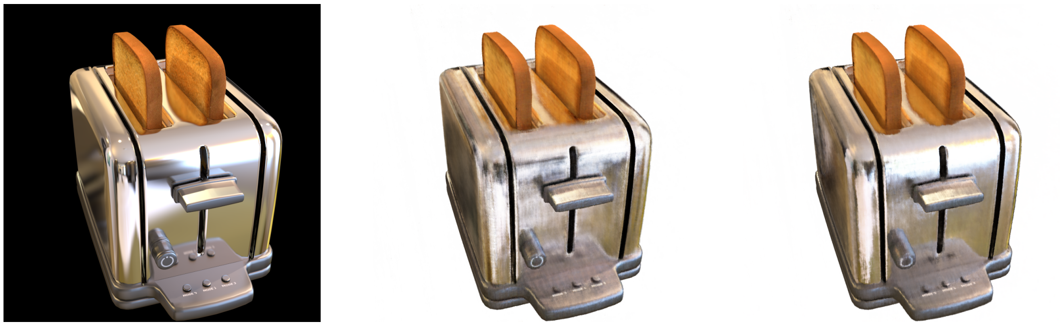
Model	*toaster*	*ball*
Ref-NeRF (from the paper)	0.922	0.995
Ref-NeRF (mine)	0.865	-
SABRe-NeRF	0.881	0.976

(b) Per scene test set SSIMs↑.

Model	*toaster*	*ball*
Ref-NeRF (from the paper)	42.87	1.548
Ref-NeRF (mine)	49.80	-
SABRe-NeRF	61.93	93.27

(c) Per scene test set MAEs↓.

Table 1: Test set evaluations. The arrow demonstrates if smaller or larger is better.



(a) The ground truth toaster image.

(b) The toaster scene rendered with Ref-NeRF.

(c) The toaster scene rendered by SABRe-NeRF.

Figure 5: The same image of the shiny blender toaster scene rendered by both the baseline Ref-NeRF model and model from this paper. Notice the crisper reflection of the toast on the top of the toaster and the matching intensity of the specular highlights along the front edge of the toaster in the image rendered by SABRe-NeRF compared to Ref-NeRF.

for full training of the model, so extending the number of training iterations to match the original papers training could lead to better understanding of the true limitations of this paper.

To view the code or reproduce the results yourself, follow the instructions at https://github.com/silas-mohr/multinerf.

Acknowledgments and Disclosure of Funding

I would like to acknowledge Professor Felix Heide for all of his wonderful and informative lectures that I learned so much from this year. I would also like to thank the course's TA, Fangyin Wei, for her invaluable advice and direction on this project. Without her help, I would not have been able to finish this paper.

References

Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. 2021. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. *CoRR*, abs/2103.13415.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. 2018. JAX: composable transformations of Python+NumPy programs.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. 2023. Flax: A neural network library and ecosystem for JAX.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. Nerf: Representing scenes as neural radiance fields for view synthesis. *CoRR*, abs/2003.08934.

Zhe Jun Tang, Tat-Jen Cham, and Haiyu Zhao. 2023. Able-nerf: Attention-based rendering with learnable embeddings for neural radiance field.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Dor Verbin, Peter Hedman, Ben Mildenhall, Todd E. Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. 2021. Ref-nerf: Structured view-dependent appearance for neural radiance fields. *CoRR*, abs/2112.03907.

Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P. Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas A. Funkhouser. 2021. Ibrnet: Learning multi-view image-based rendering. *CoRR*, abs/2102.13090.