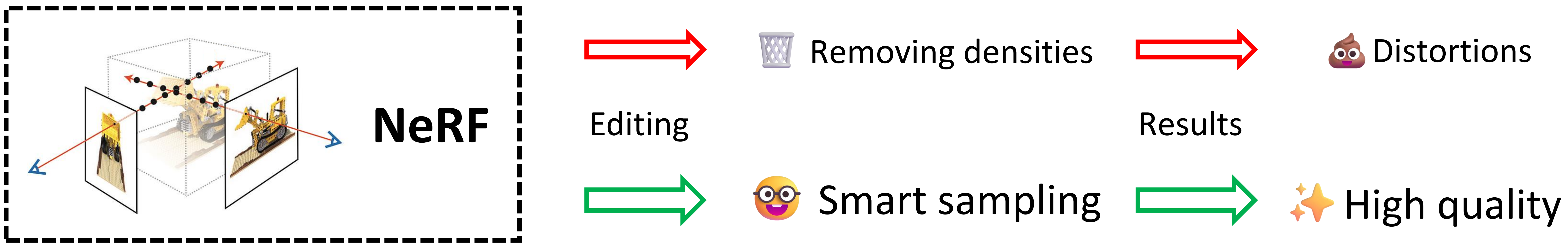
SMART SAMPLING FOR OBJECT REMOVAL OPERATIONS IN NEURAL RADIANCE FIELDS

Mikołaj Zieliński[1,2], Dominik Belter[2], Peyman Moghadam[1,3]

OBJECTIVES



Over the past few years, **Neural Radiance Fields (NeRFs)** have gained considerable popularity due to their ability to **generate accurate spatial representations** based on just a few images of the desired scene. However, one drawback of NeRF is the **inability to implicitly modify the generated representation** once training has been done. Many different approaches have been proposed that attempt to edit NeRFs explicitly. The method proposed in this paper belongs to a group of **editing NeRFs by modifying the individual densities** generated by the NeRFs. Those approaches allow for fast object removal but often generate neural field distortions behind the removed objects. We assumed that the source of this issue might be an insufficient number of samples of the rendered region. To mitigate this problem **we propose a sampling strategy that allows for object removal without compromising rendering quality and computation time**. Our method ensures the elimination of the distortion regions and provides accurate sampling of the region of interest. Moreover, our procedure leads to an accurate geometry representation that is essential for robotic applications.

METHODOLOGY

Following NeRF paper, we encode a scene as an $\boldsymbol{MLP_\Theta}$ **that predicts color c of a point and density σ**, given a 5-dimensional input containing x, y, z and ray direction angles $\theta, \varphi, F(x, y, z, \theta, \varphi) = (\sigma, c)$. Point parameters in space can be calculated from ray equation r,

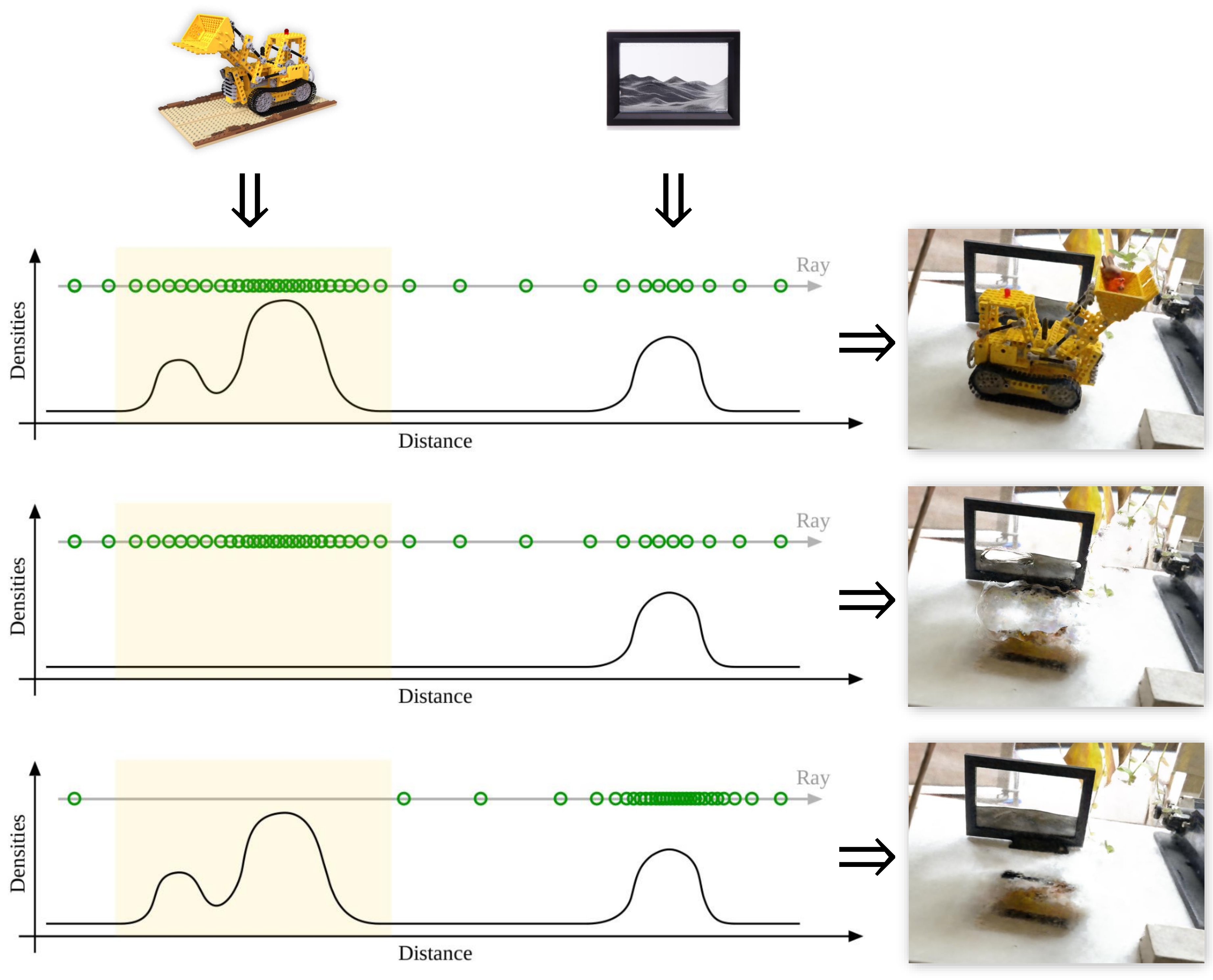
$$r(t) = \mathbf{o} + \mathbf{d}t$$

After querying network F for multiple points obtained from the sampler for each ray, a **novel view of a scene can be rendered using the following equation**,

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(r(t))c(r(t), d)dt$$

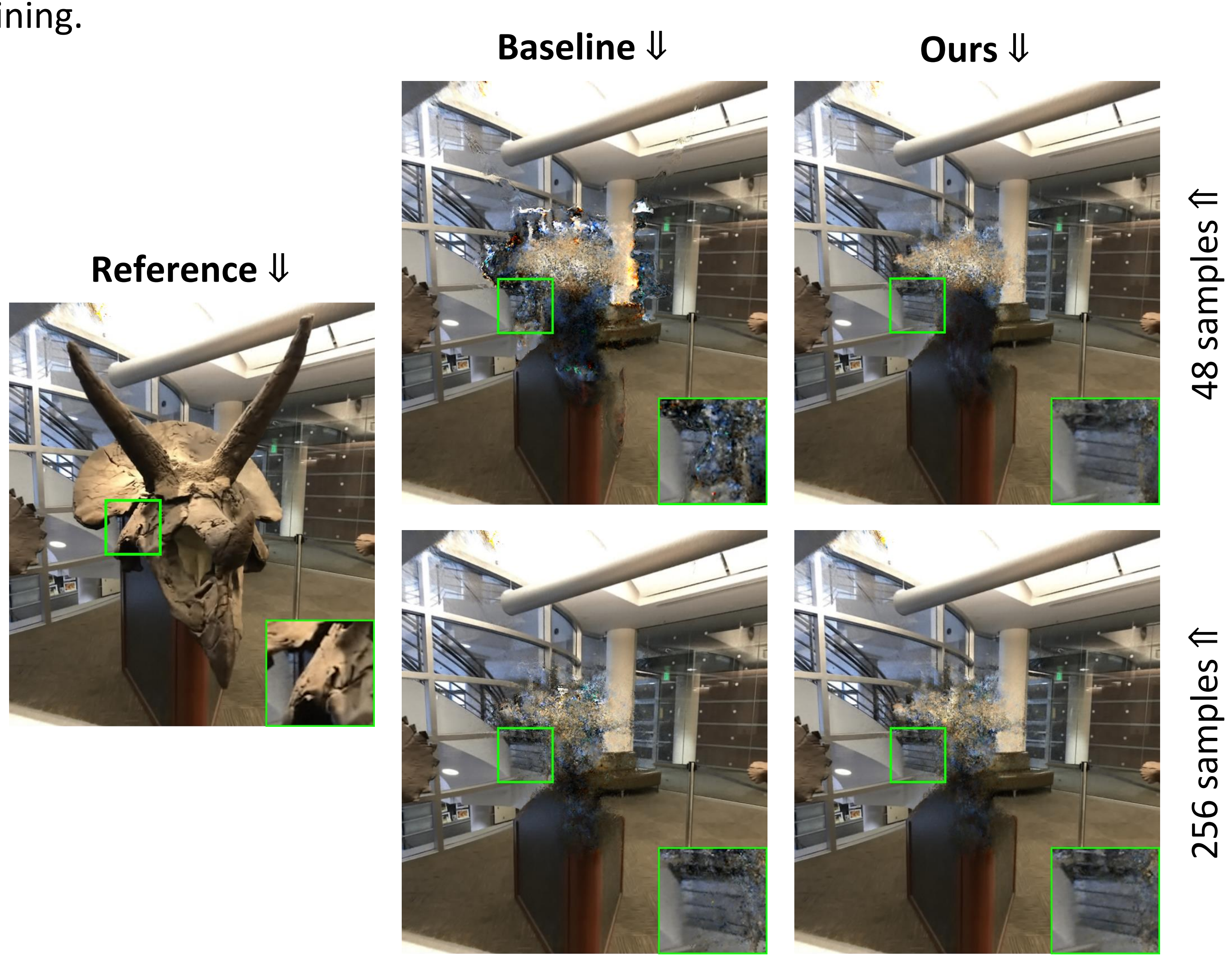
If ray is hitting the surface *and* we are on the surface *output* the color c

Most of the NeRF editing methods that allow for removing objects from this representation are **suppressing the density values**. Therefore according to the above equation points belonging to the removed object **won't be taken into consideration during the rendering**. This means that the number of points contributing to the reconstructed regions will be reduced. We have assumed that this is the source of distortions in such editing approach. To mitigate this problem we propose instead of removing unwanted densities, to **not sample from the region occupied by the object** we want to remove. This will cause all the samples to shift to the regions we want to reconstruct leading towards **high quality reconstruction**. Explanation of our method is visible in the graphic below:



RESULTS

We have conducted two experiments. The first one with a number of samples equal to 48 and the second one with a number of samples equal to 256. Below figures show that **our method outperforms the baseline** when the number of samples is equal to 48 and is slightly better when the number of samples is equal to 256. The real difference between 48 and 256 samples comes in terms of **much longer network inference time and higher GPU resource usage**. Additionally, we can notice that the baseline model has regions of neural field distortions behind the removed skull, while our method addresses this drawback. Both sampling techniques still have problems with regions that were not visible at all during the training.



To determine the quality of the reconstruction of occluded regions, we used synthetic data. This allowed us to obtain ground truth images to generate **precise masks for occluded regions**. We have generated **five different scenes** with random objects on the table. Our task was to **remove objects and reconstruct the area behind them**. After training, we were able to calculate metrics such as PSNR for the whole RGB image, PSNR-M only for masked regions of RGB images, MSE for depth images, and MSE-M for depth images but only for masked regions. Our results can be seen in the table below. We can observe that **our method outperforms the baseline in all the cases** for both RGB and depth images.

Scene	PSNR [dB] ↑		PSNR-M [dB] ↑		MAE [m] ↓		MAE-M [m] ↓	
	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours
1	29.76	**34.65**	18.63	**27.39**	0.032	**0.00063**	0.550	**0.0024**
2	30.92	**32.10**	25.50	**31.21**	0.013	**0.0042**	0.130	**0.0230**
3	31.17	**32.23**	27.02	**27.72**	0.015	**0.0086**	0.054	**0.0220**
4	38.38	**40.78**	26.48	**33.87**	0.003	**0.0013**	0.098	**0.0180**
5	29.49	**29.99**	23.73	**25.25**	0.032	**0.0018**	0.400	**0.0069**
Mean	31.94	**33.95**	24.27	**29.09**	0.019	**0.0033**	0.248	**0.0145**

ACKNOWLEDGEMENTS

This work was funded by CSIRO Robotics.
The work was partially supported by the National Science Centre, Poland, under research project no UMO-2023/51/B/ST6/01646.

[1]CSIRO Robotics, Data61, Pullenvale Qld 4069, Australia.
firstname.lastname@csiro.au

[2]Poznan University of Technology, Institute of Robotics and Machine Intelligence, ul. Piotrowo 3A, Poznań 60-965, Poland.
dominik.belter@put.poznan.pl, mikolaj.zielinski@doctorate.put.poznan.pl

[3]School of Electrical Engineering and Robotics, Queensland University of Technology (QUT), Brisbane, Australia.
firstname.lastname@qut.edu.au