JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

FreeNeRF: Improving Few-shot Neural Rendering with Free Frequency Regularization

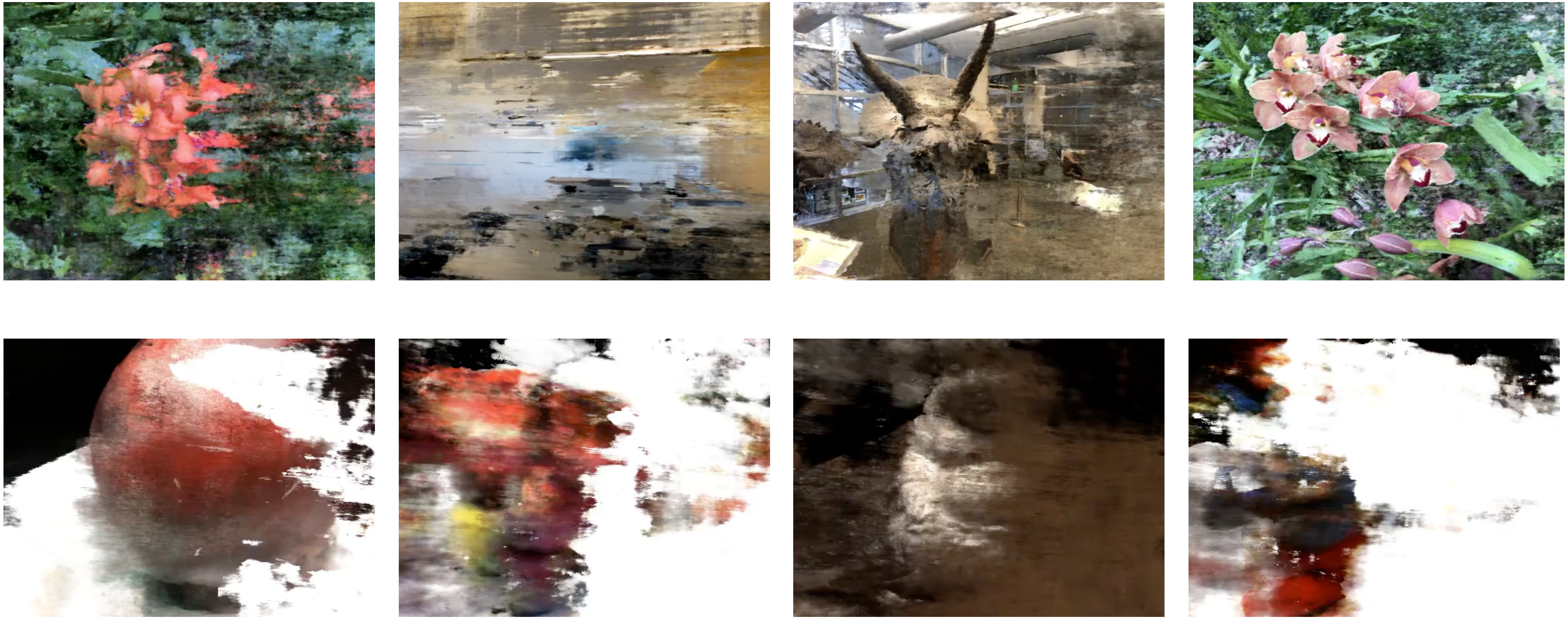
Jiawei Yang
UCLA

Marco Pavone
Nvidia Research, Stanford University

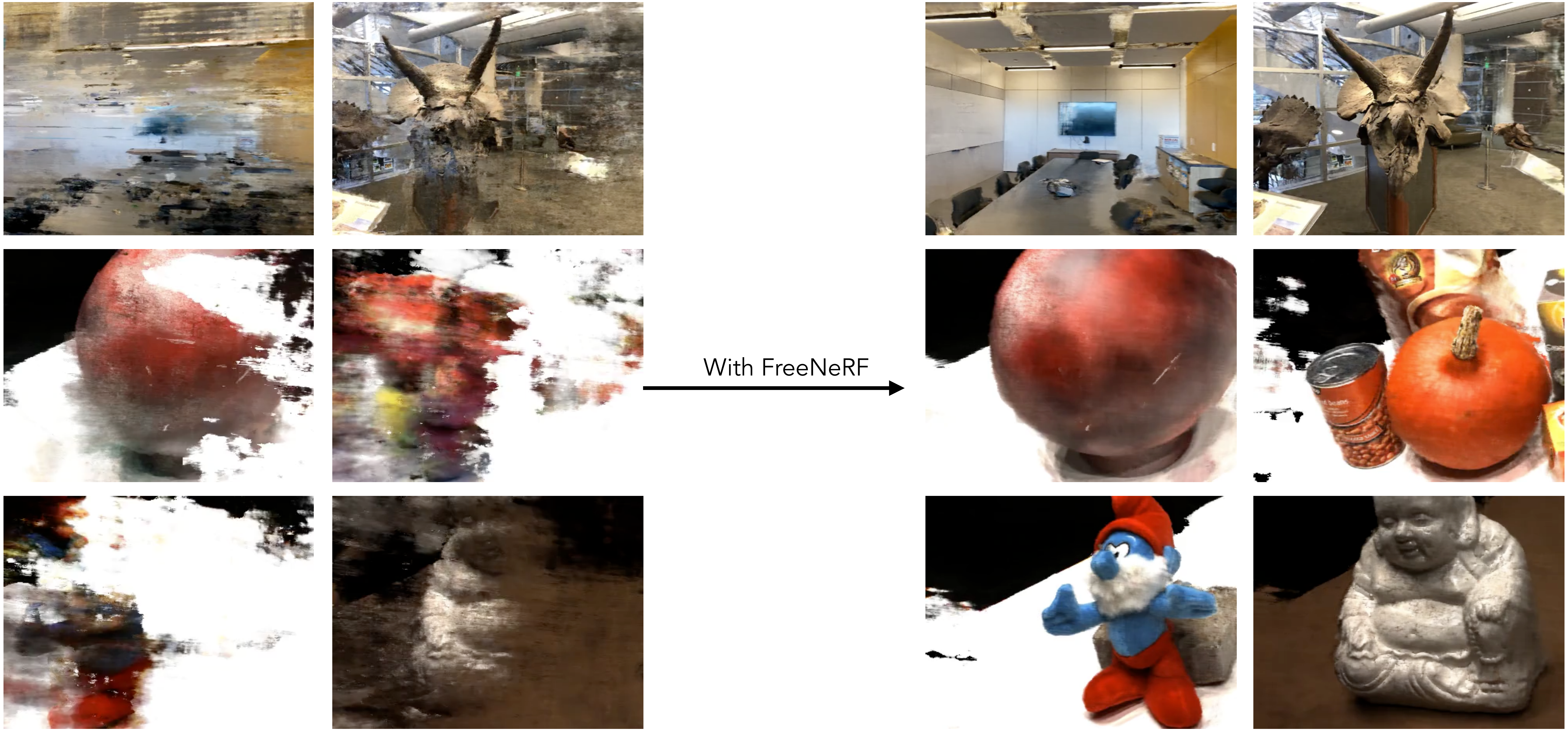
Yue Wang
Nvidia Research

Paper tag: WED-AM-003

NeRF struggles at novel view synthesis from sparse views



FreeNeRF improves few-shot neural rendering with a few lines of code



FreeNeRF stabilizes few-shot NeRF's training by frequency regularization.

Positional encoding

$$\gamma(p) = \left(\sin\left(2^0\pi p\right), \cos\left(2^0\pi p\right), \ldots, \sin\left(2^{L-1}\pi p\right), \cos\left(2^{L-1}\pi p\right)\right)$$

Visible inputs to NeRF

Position encoded bits

Bits with frequency 2^0 (low) Bits with frequency 2^{L-1} (High)

Training iterations:

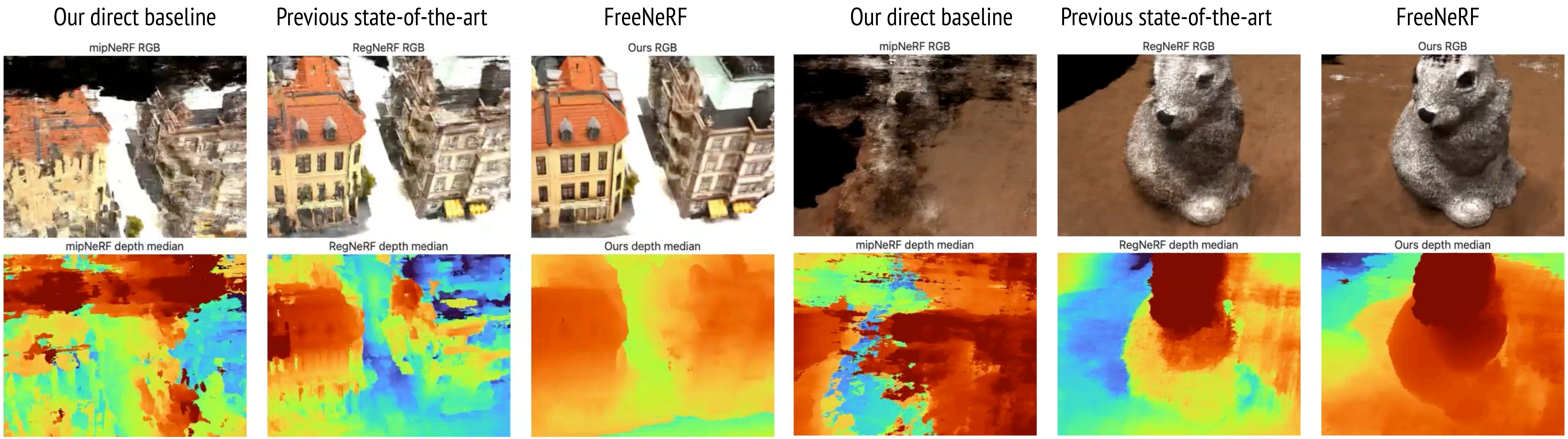
FreeNeRF removes few-shot NeRF's by occlusion regularization.

FreeNeRF without occlusion regularization

+ occlusion regularization

FreeNeRF with occlusion regularization

FreeNeRF achieves new state-of-the-art performance with minimal overhead.



3-shot	DTU obj. PSNR	LLFF PSNR	Training Time
MipNeRF	9.10	16.11	1x
RegNeRF	18.50	18.84	1.69~1.98x
Ours	19.92	19.63	$\boxed{1.02\text{~}1.04x}$

8-shot	Blender PSNR	Training Time
MipNeRF	13.93	1x
DietNeRF	22.50	2.8x
Ours	24.26	$\boxed{1.02x}$

What makes FreeNeRF different?

FreeNeRF

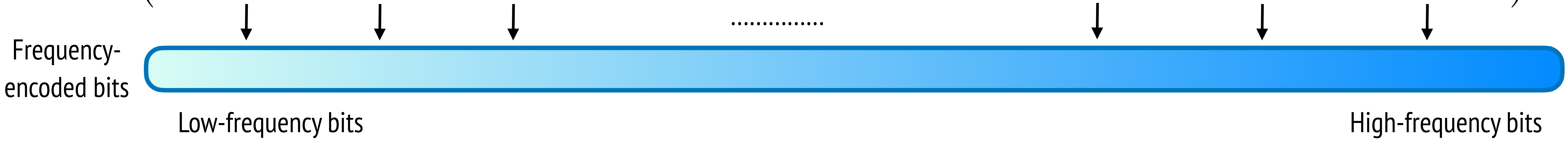
- Doesn't require any pre-training / pre-trained models.
- Doesn't rely on additional depth information.
- Easy to implement (a few lines of code)
- Fast to run. It only takes about 1.02x training time compared to plain NeRFs.

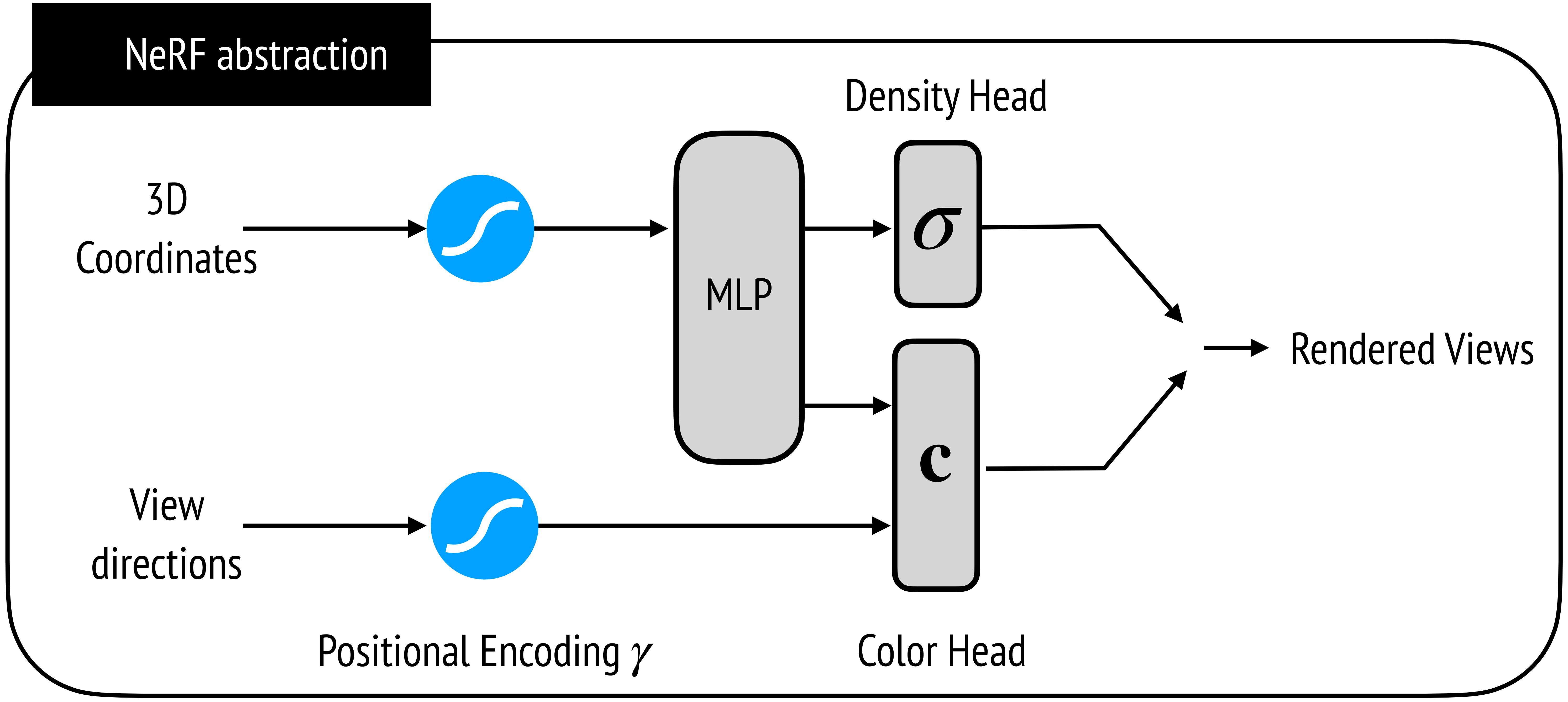
PixelNeRF[1]	DieNeRF[3]	DSNeRF[5]
MVSNeRF[2]	RegNeRF[4]	...
...	...	
Use costly pre-training & extra CNN models	Use pretrained models for regularization	Use depth priori

1. Yu, Alex, et al. "pixelnerf: Neural radiance fields from one or few images." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.
2. Chen, Anpei, et al. "Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.
3. Jain, Ajay, Matthew Tancik, and Pieter Abbeel. "Putting nerf on a diet: Semantically consistent few-shot view synthesis." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.
4. Niemeyer, Michael, et al. "Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.
5. Deng, Kangle, et al. "Depth-supervised nerf: Fewer views and faster training for free." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

The de-facto standard in NeRF: position encoding.

$$\gamma(p) = \left(\sin\left(2^0 \pi p\right), \cos\left(2^0 \pi p\right), \sin\left(2^1 \pi p\right), \cos\left(2^1 \pi p\right), \ldots, \sin\left(2^{L-2} \pi p\right), \cos\left(2^{L-2} \pi p\right), \sin\left(2^{L-1} \pi p\right), \cos\left(2^{L-1} \pi p\right)\right)$$

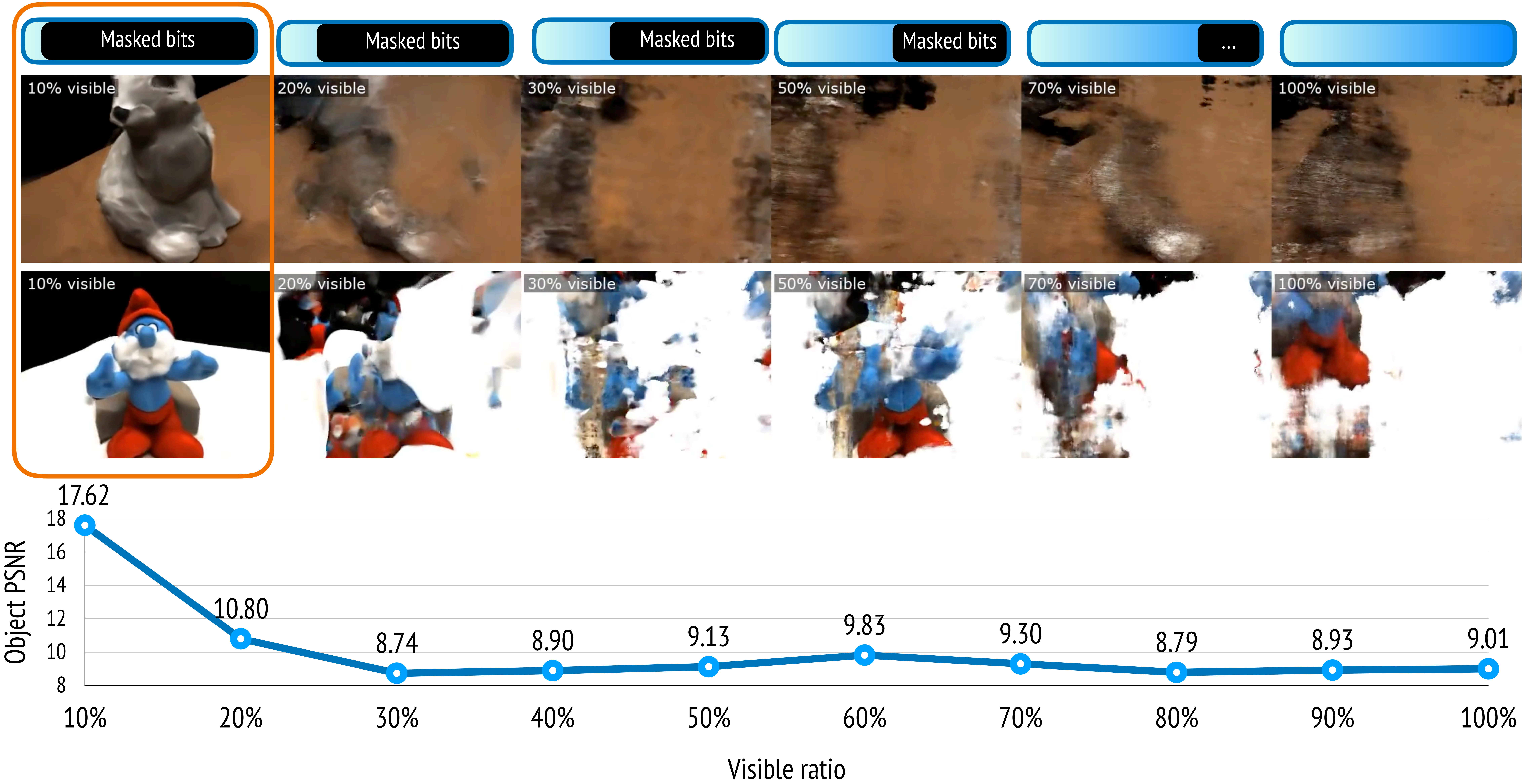




- High-frequency mapping enables **faster** convergence on high-frequency components[6].
- Do we want this property in few-shot NeRF's learning?
  - **No!**

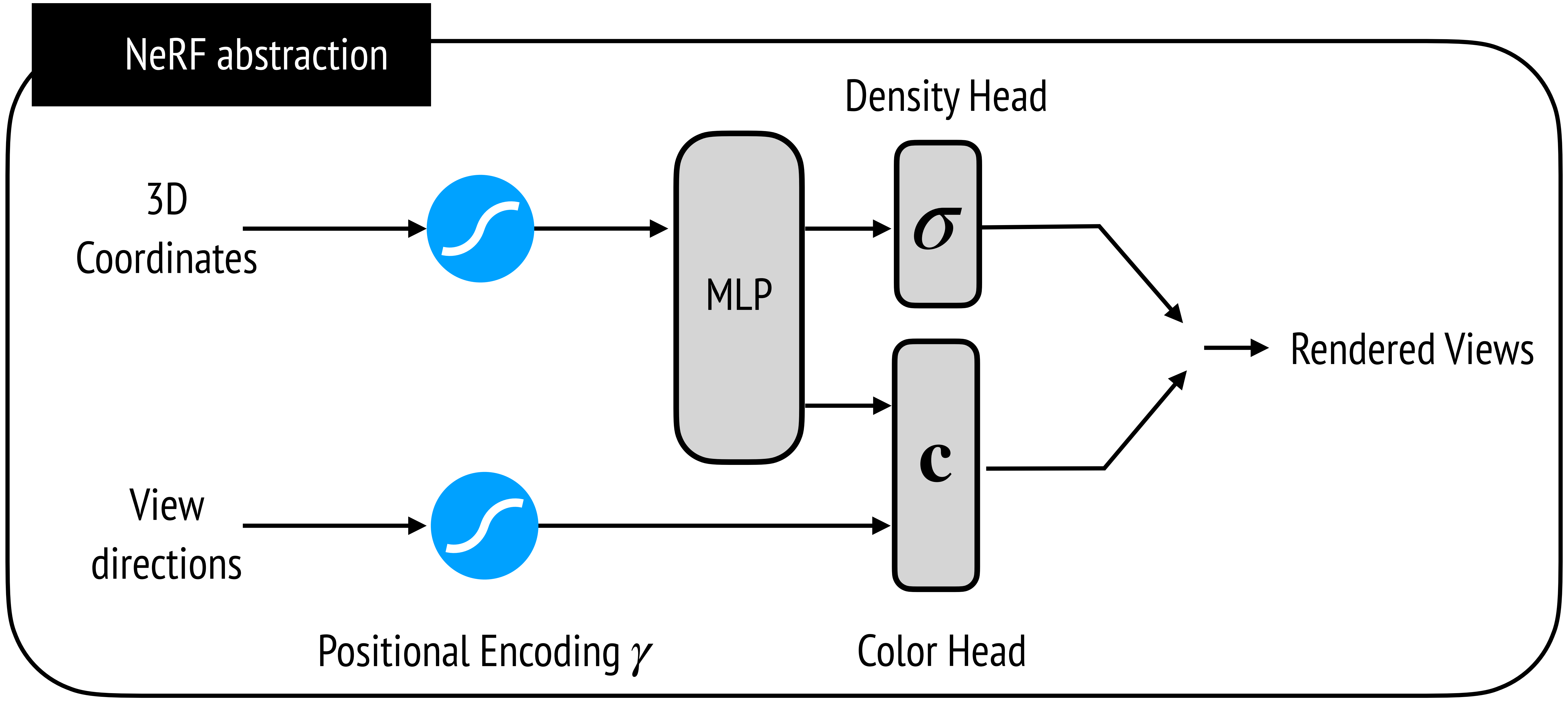
6. Tancik, Matthew, et al. "Fourier features let networks learn high frequency functions in low dimensional domains." Advances in Neural Information Processing Systems 33 (2020): 7537-7547.

Pilot Study: Low-frequency-only-inputs work surprisingly well!



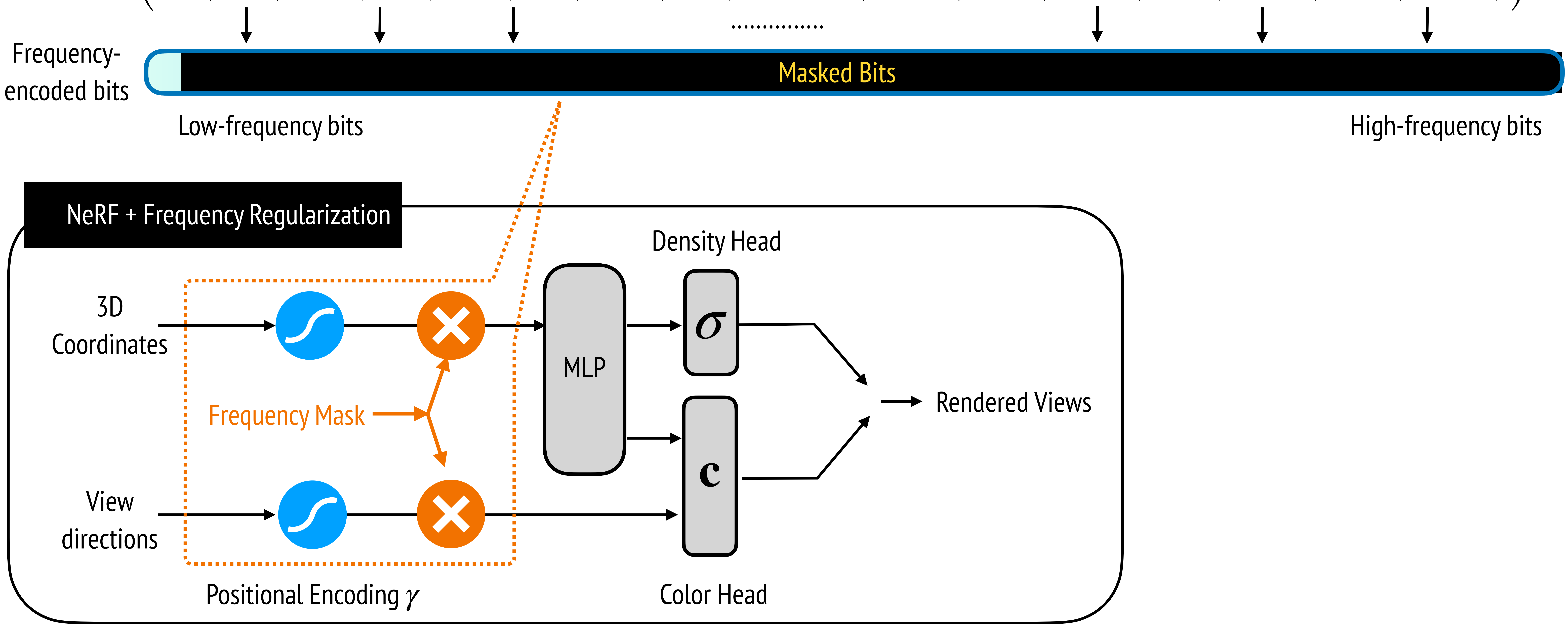
Frequency Regularization

$$\gamma(p) = \left(\sin\left(2^0 \pi p\right), \cos\left(2^0 \pi p\right), \sin\left(2^1 \pi p\right), \cos\left(2^1 \pi p\right), \ldots, \sin\left(2^{L-2} \pi p\right), \cos\left(2^{L-2} \pi p\right), \sin\left(2^{L-1} \pi p\right), \cos\left(2^{L-1} \pi p\right)\right)$$



Frequency Regularization

$$\gamma(p) = \left(\sin\left(2^0\pi p\right), \cos\left(2^0\pi p\right), \sin\left(2^1\pi p\right), \cos\left(2^1\pi p\right), \ldots, \sin\left(2^{L-2}\pi p\right), \cos\left(2^{L-2}\pi p\right), \sin\left(2^{L-1}\pi p\right), \cos\left(2^{L-1}\pi p\right)\right)$$



Motivation: Frequency Matters!

Training iterations:

Scheduled frequency mask

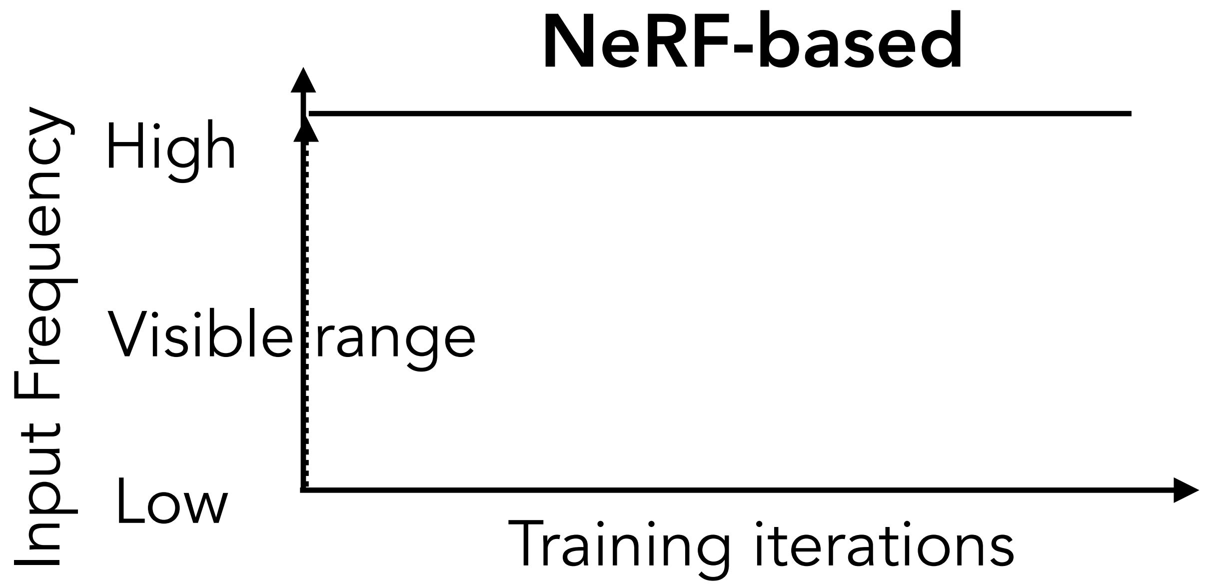
Masked bits

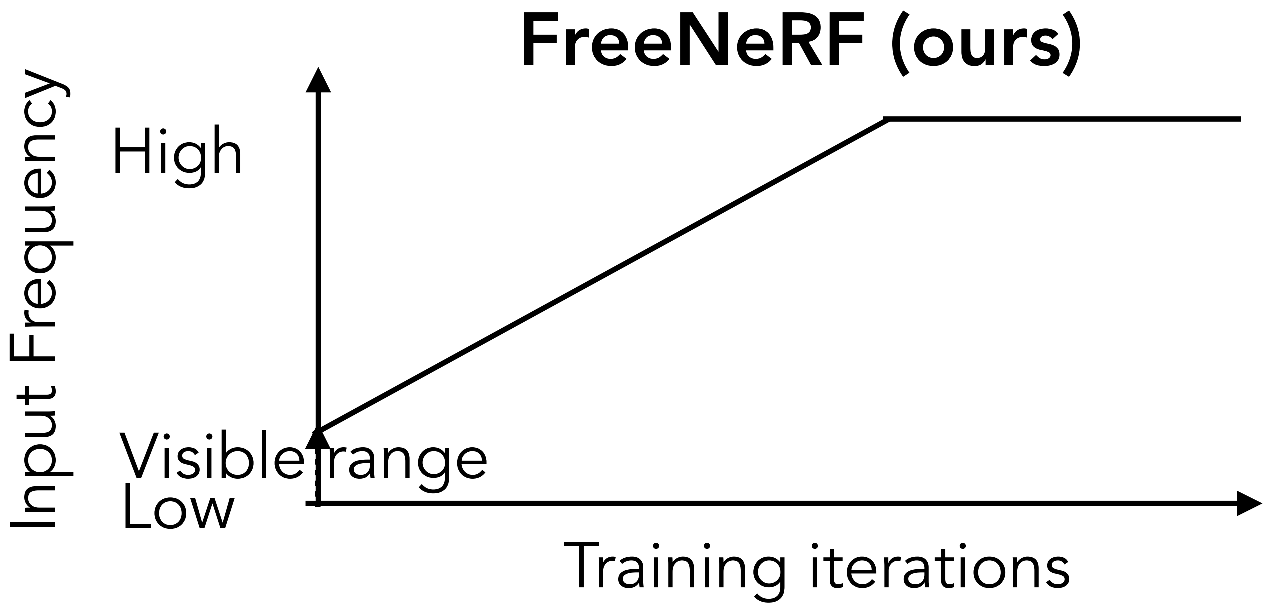
Bits with lower frequency

Bits with higher frequency

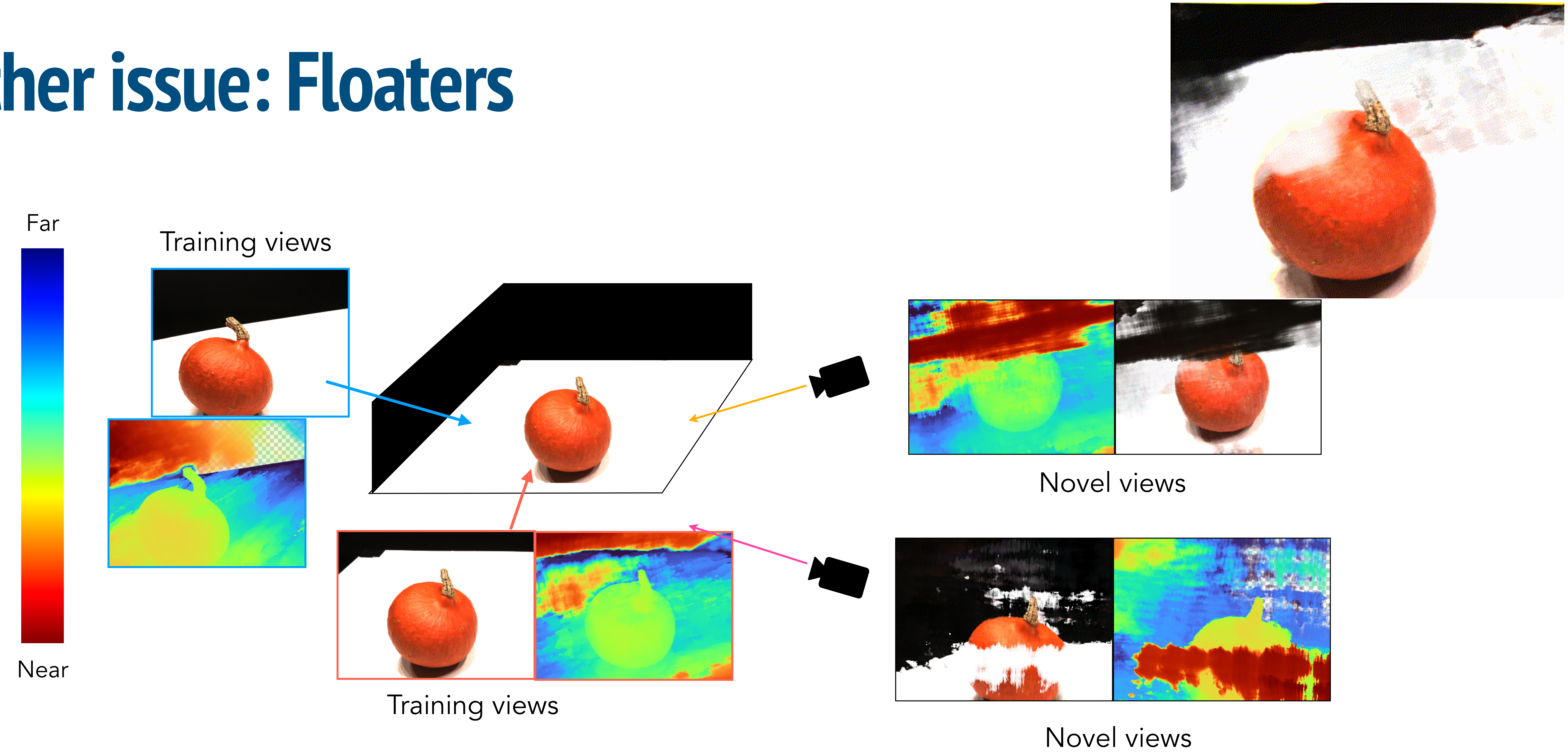
Motivation: Frequency Matters!

Scheduled frequency mask

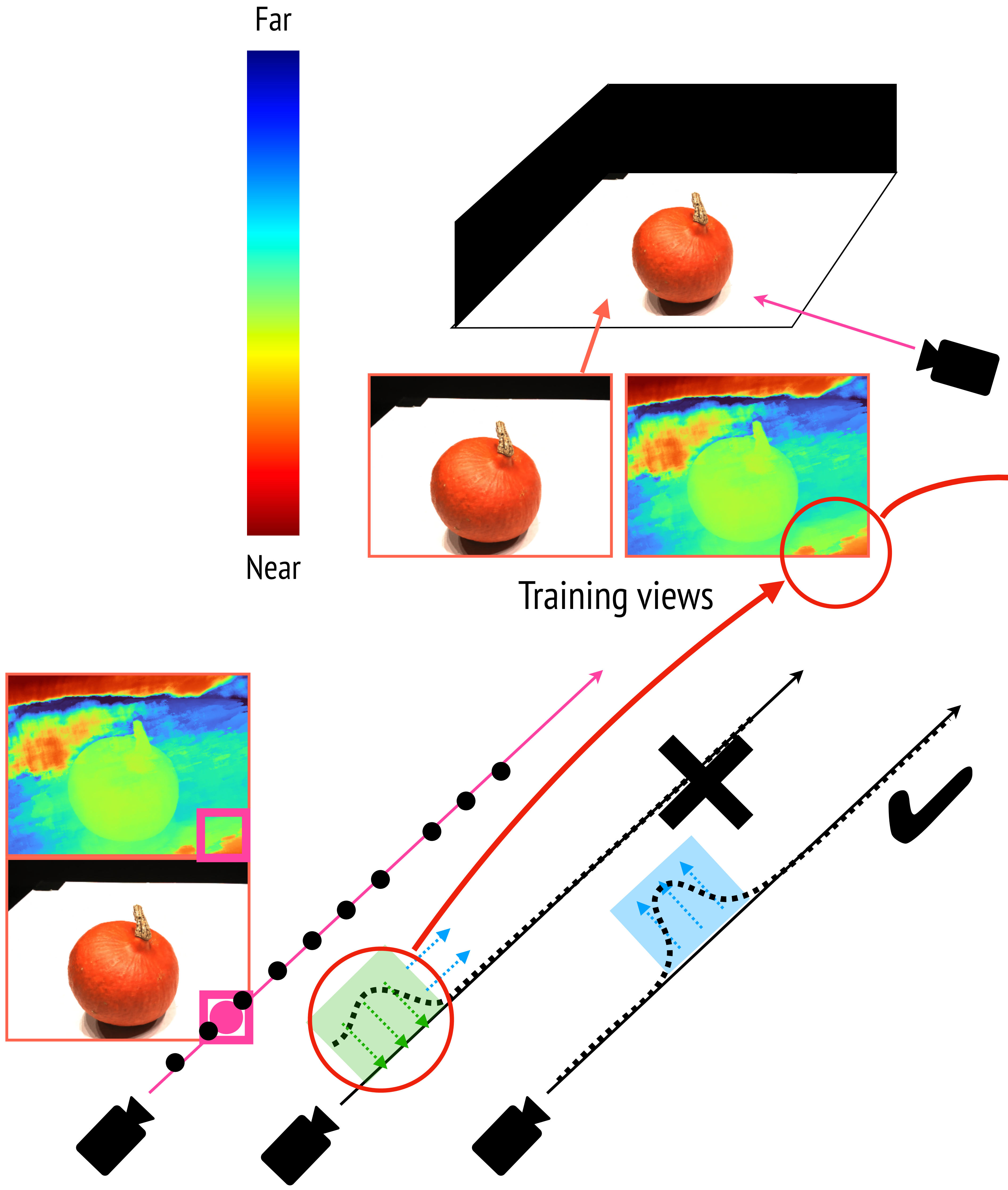


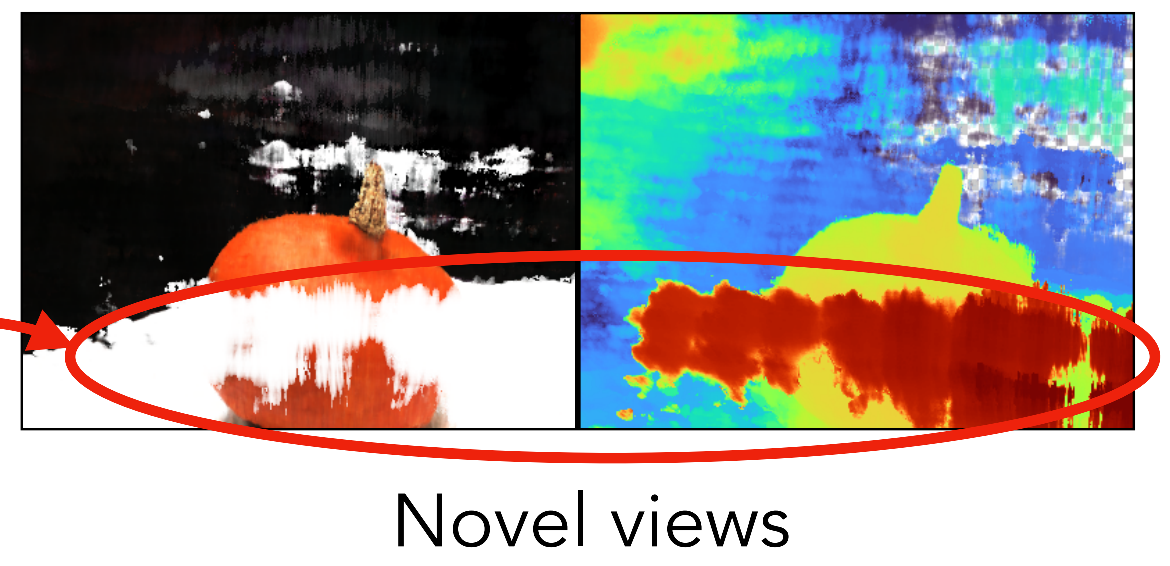


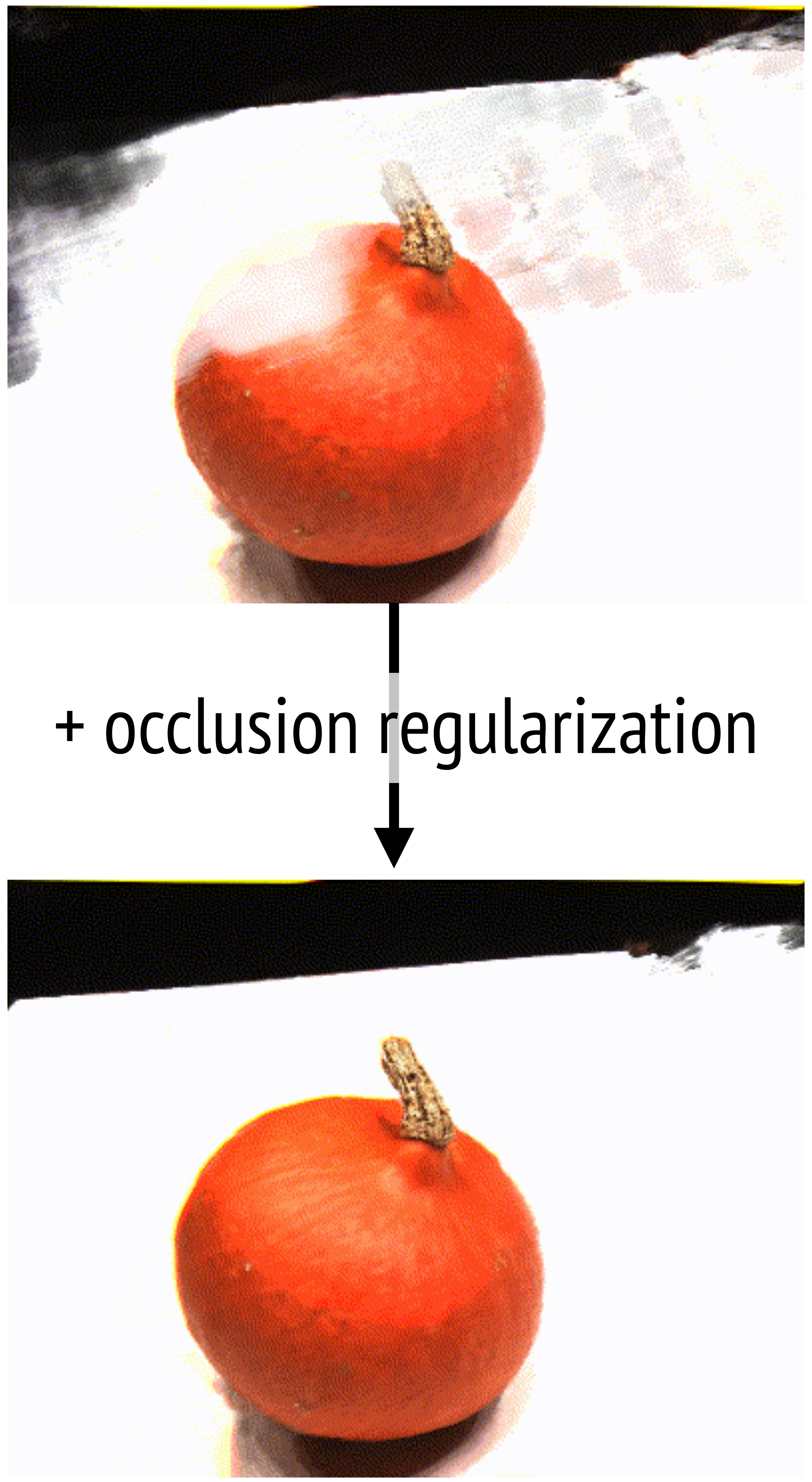
Another issue: Floaters



Motivation: Push floaters away





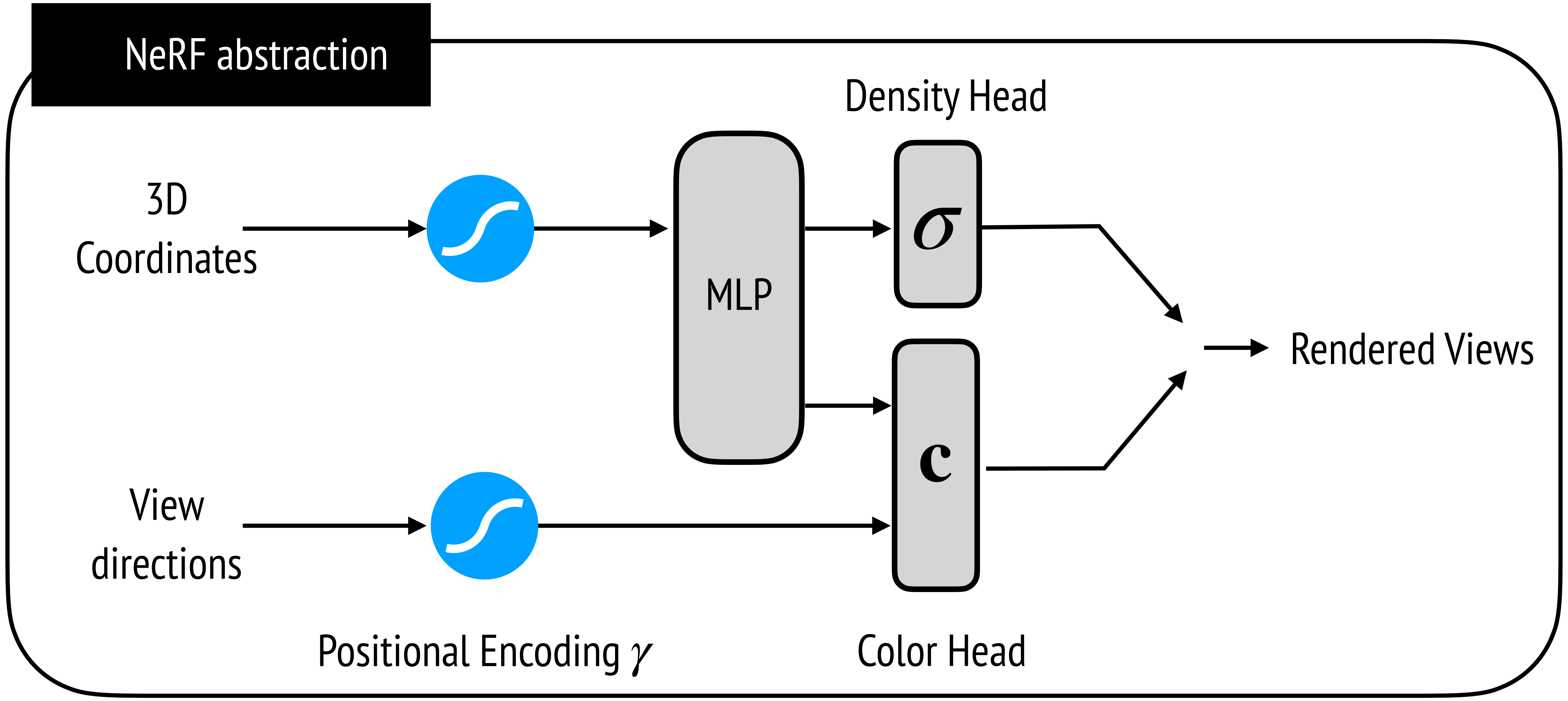


(1) Push the density of floaters in the near-camera regions (m_k) to zeros

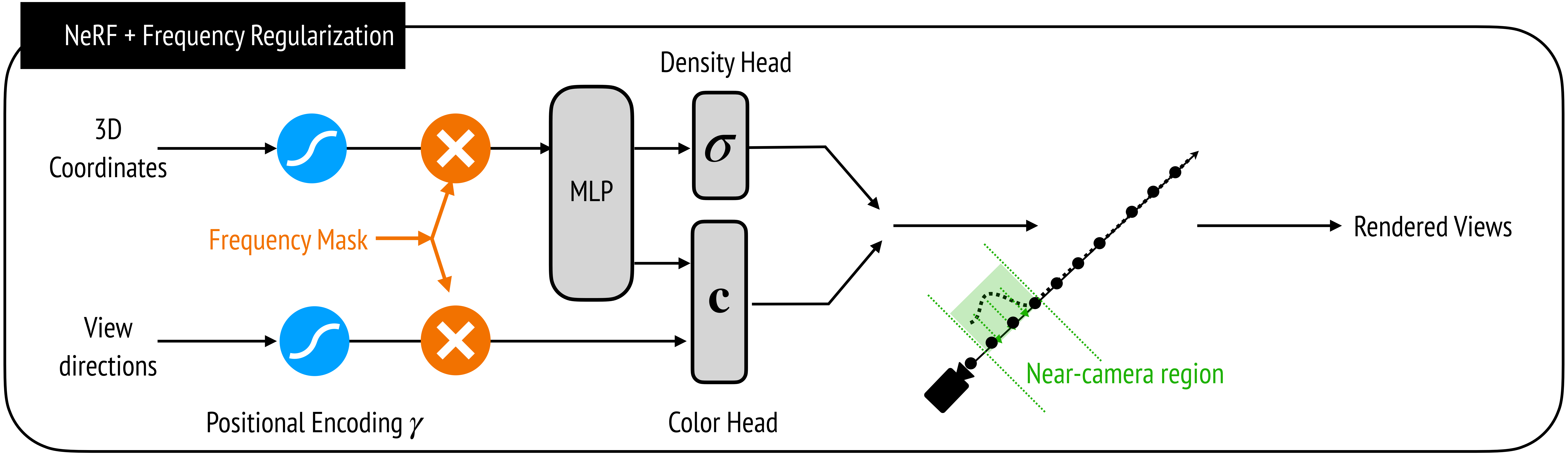
(2) Models learn to explain this area in a farther place

$$\mathscr{L}_{occ} = \sum_{k=1}^{K} \sigma_k \cdot m_k / K$$

FreeNeRF Summary

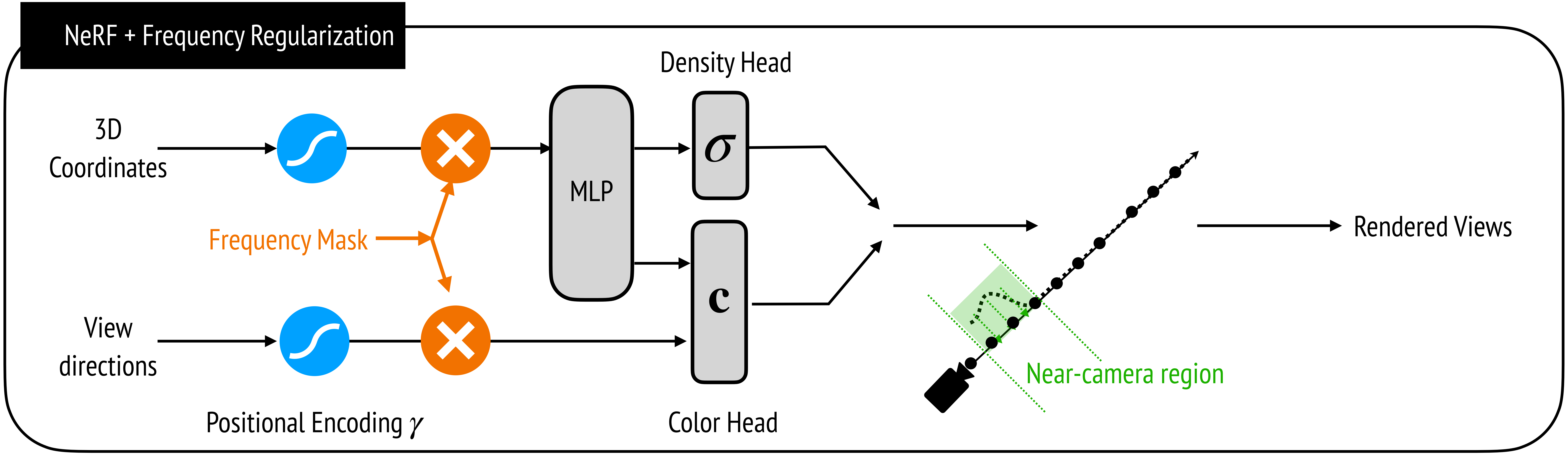


FreeNeRF Summary



- We use frequency regularization to stabilize NeRF's training.
- We use occlusion regularization to address floater issues.

Easy implementation!



- We use frequency regularization to stabilize NeRF's training.

```
freq_mask = ones_like(pos_enc)
freq_mask[:, int(t/T*L)+3:]  = 0
NeRF_Inputs = pos_enc * freq_mask
```

Or simply as

```
pos_enc[:, int(t/T*L)+3:] = 0
```

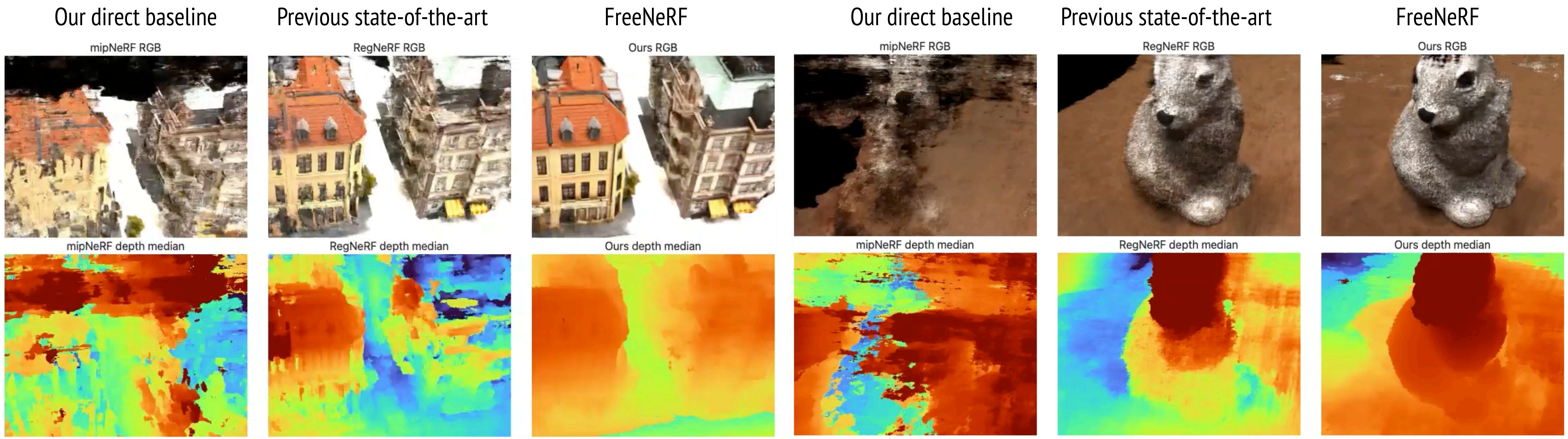
- We use occlusion regularization to address floater issues.

```
occ_mask = ones_like(queried_density)
occ_mask[:, : occ_reg_range, :]  = 1.0
occ_loss = (queried_density * occ_mask).mean()
```

Or simply as

```
occ_loss = queried_density[:, : occ_reg_range, :].mean()
```

FreeNeRF achieves new state-of-the-art performance with minimal overhead



3-shot	DTU obj. PSNR	LLFF PSNR	Training Time
MipNeRF	9.10	16.11	1x
RegNeRF	18.50	18.84	1.69~1.98x
Ours	19.92	19.63	$\boxed{1.02\text{~}1.04x}$

8-shot	Blender PSNR	Training Time
MipNeRF	13.93	1x
DietNeRF	22.50	2.8x
Ours	24.26	$\boxed{1.02x}$

FreeNeRF: Improving Few-shot Neural Rendering with Free Frequency Regularization

- We use frequency regularization to stabilize NeRF's training.
- We use occlusion regularization to address floater issues.
- Our simply-designed FreeNeRF achieve new state-of-the-art performance on three few-shot benchmarking datasets!
- Please refer to our project page and open-sourced code for more details:
  - Projection page: https://jiawei-yang.github.io/FreeNeRF/
  - Code Page: https://github.com/Jiawei-Yang/FreeNeRF

Project Page

Code Page

Thanks for watching!