ICLR 2024 Submission

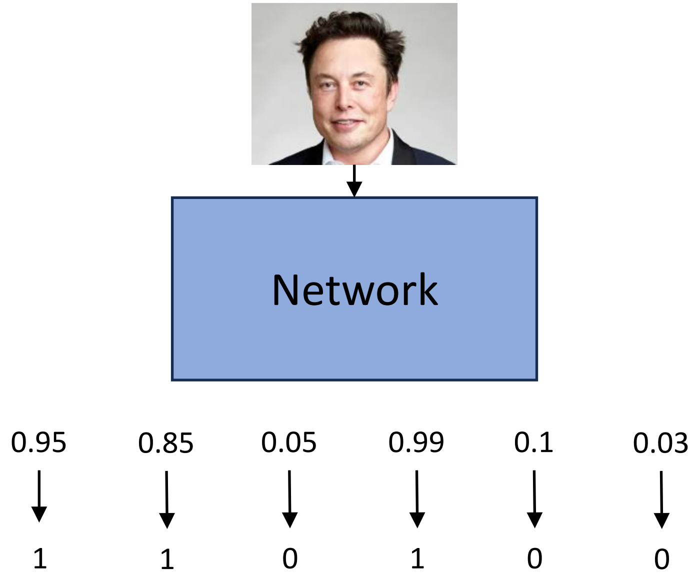
Neural Field Classifiers via Target Encoding and Classification Loss

Anonymous authors

Efficient Learning Lab@POSTECH

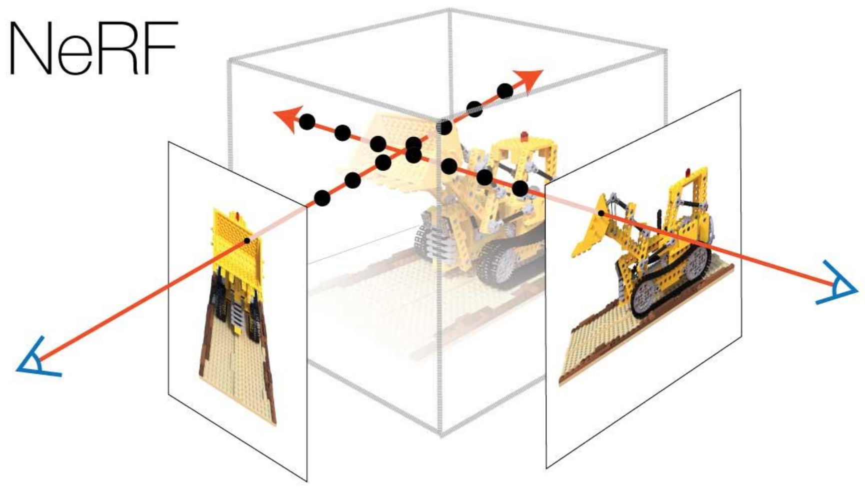
Junwon Seo

Recap: LEARNING LABEL ENCODINGS FOR DEEP REGRESSION



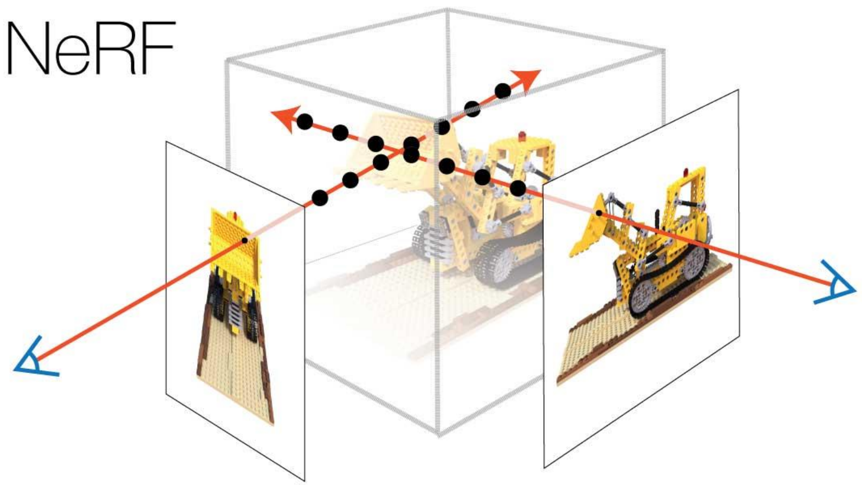
- Example
  - The age of 52 would be 110100 using binary conversion as the encoder. (Training phase)
  - 110100 would be converted to real-value prediction using a decoding function (decimal convert) (Inference phase)

Neural Field



- Neural field methods emerge as promising methods for parameterizing **a field**
  - that has **a target value** for **each point in space and time**
- Try to predict some coordinate-based continuous target values
  - RGB for Neural Radiance Field

NeRF Basics



$$\hat{\boldsymbol{C}}(\boldsymbol{r}) = \int_{t_n}^{t_f} T(t)\sigma(t)\boldsymbol{c}(t)dt$$

$$\text{where } T(t) = \exp(-\textstyle\int_{t_n}^{t} \sigma(s)ds)$$

$$L(\Theta) = \frac{1}{\|\mathcal{R}\|} \sum_{\boldsymbol{r}\in\mathcal{R}} \|\hat{\boldsymbol{C}}(\boldsymbol{r}) - \boldsymbol{C}(\boldsymbol{r})\|_2^2$$

- NeRF **regresses** from a single 5D representation (x, y, z, θ, ϕ) to (σ, c)
  - 3D coordinates $x =$ (x, y, z) / viewing directions d = (θ, ϕ)
  - Volume density σ and view-dependent color c =(r, g, b)

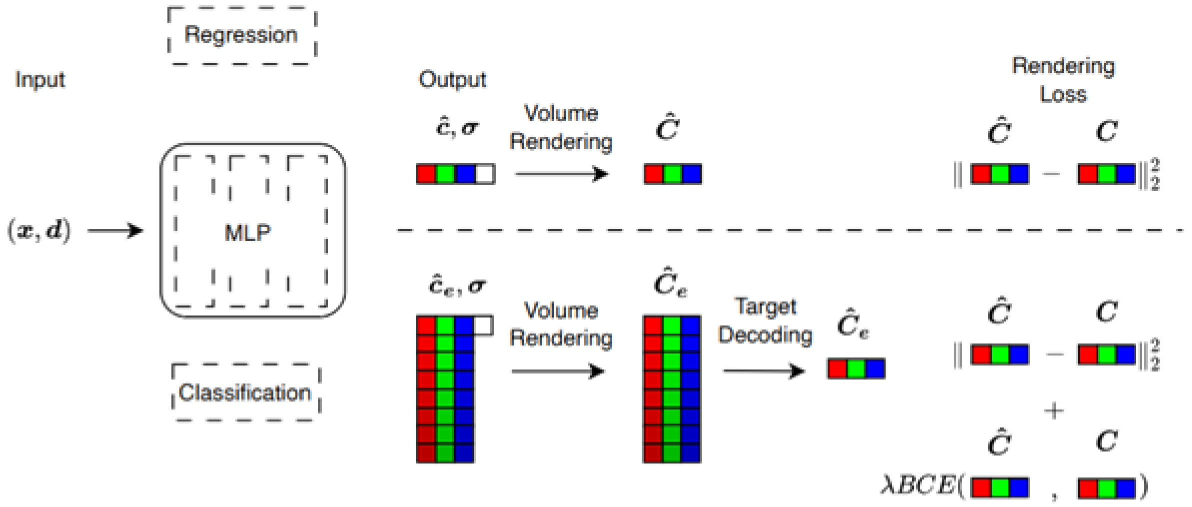
Motivation

Are **regression formulations** really better than **classification formulations** for neural field methods?

Motivation

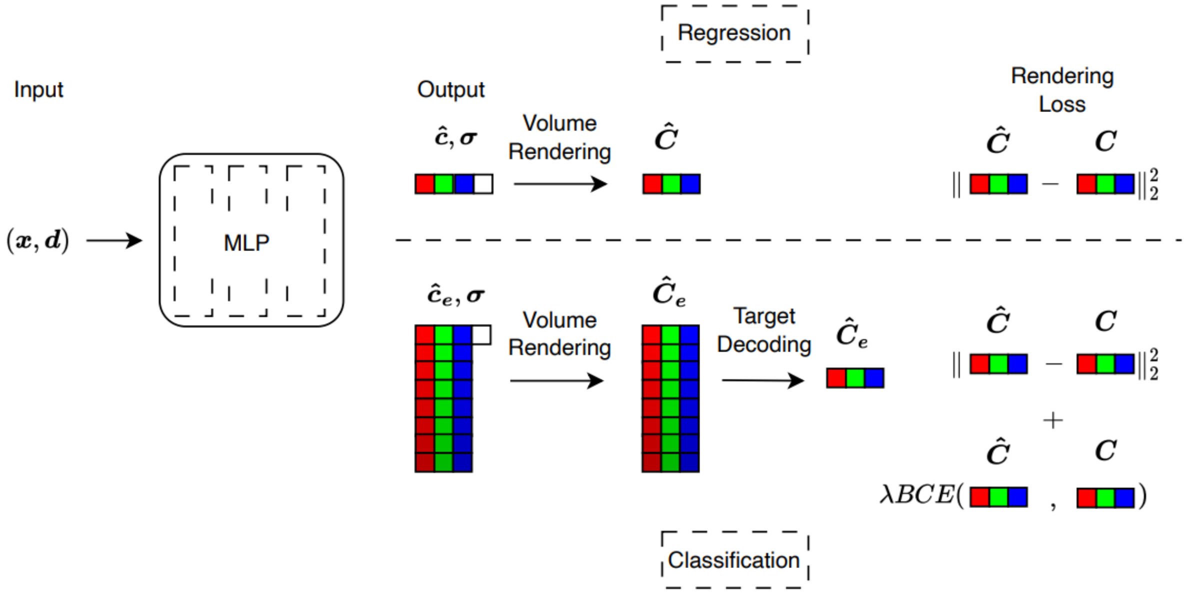
- People naturally formulate neural fields as regression models
  - Because the targets are continuous values
- The authors claimed that there exist overlooked pitfalls
  - Each data point has its own ground-truth label in classical supervised learning methods
  - NeRF output N points’ predictions per pixel.
  - Supervision signals for NeRF are obviously very weak and insufficient

Methodology – Target Encoding and Decoding



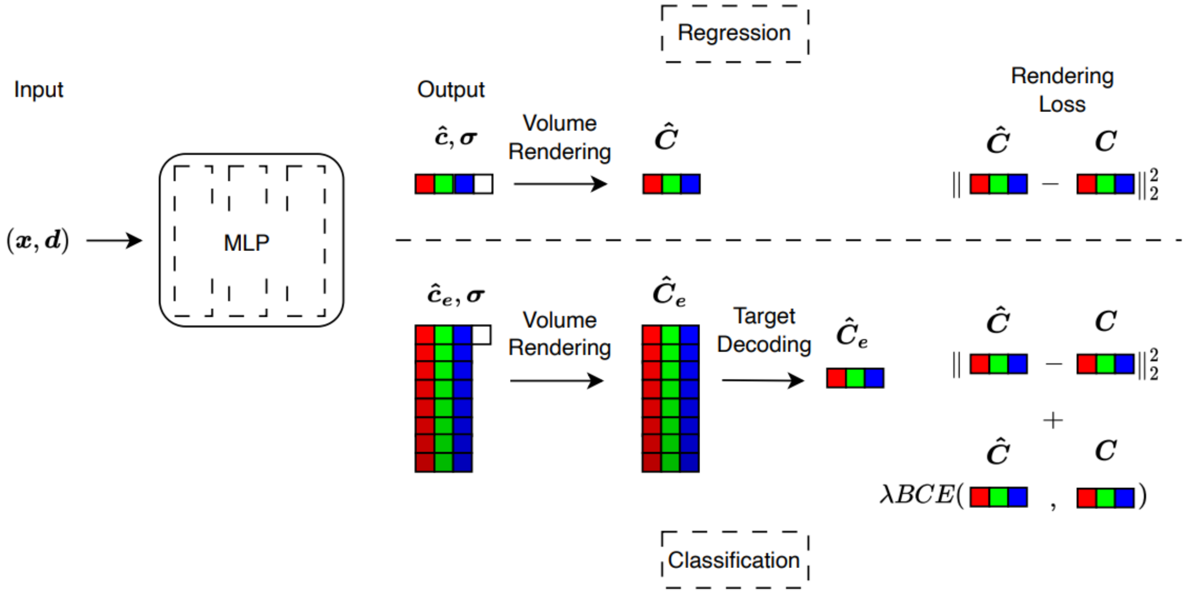
- Naïve target encoding rule is one-hot encoding
  - Directly use y as the class label (i.e. 256-class classification)
    - The number of logits increases to 768 from (computational cost ↑)
    - Ignore the relevant information carried by the classes
      - Suppose ground-truth label of a sample is 0
      - A model predicts 1 and another model predicts 255
        - Their loss will be equally high

Methodology – Target Encoding and Decoding



- Fortunately, the authors discover that the classical binary-number system can work well
  - For example, y = BinaryEncod(203) = [1, 1, 0, 0, 1, 0, 1, 1]

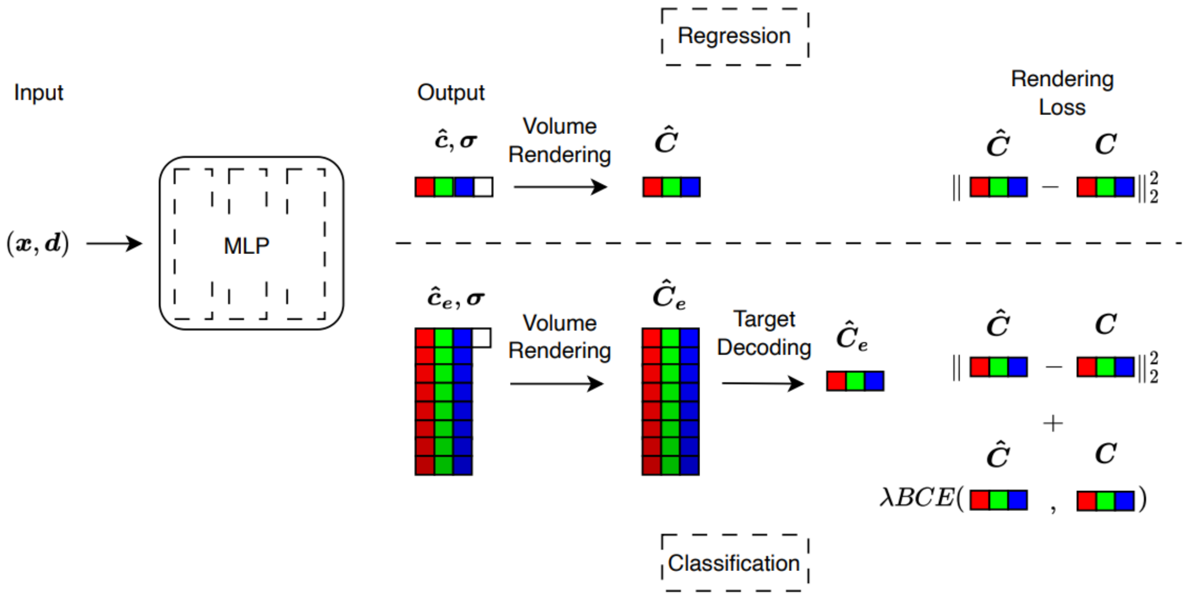
Methodology – Classification Loss



- Classification Loss
  - 2^{j-1} assigns a higher weight to the class with a higher place value.

$$l_{\mathrm{b}}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{255} \sum_{j=1}^{8} 2^{j-1} \, \mathrm{BCE}(\hat{\boldsymbol{y}}^{(j)}, \boldsymbol{y}^{(j)})$$

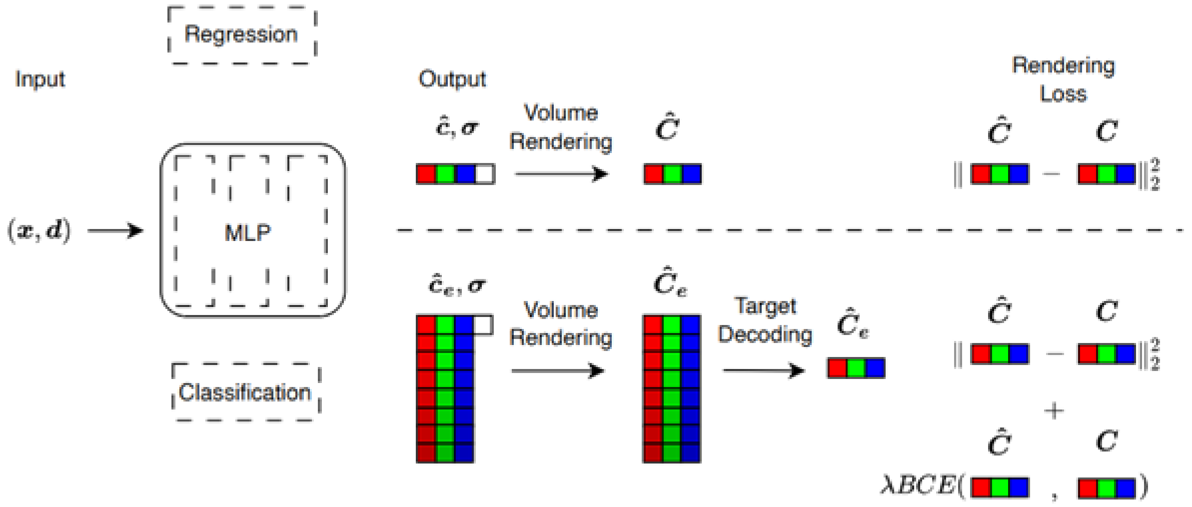
Methodology – Classification Loss



- Alternative choice is..
  - Decode the predict probability $\hat{y}$ back into a continuous value

$$l_c(\hat{C}, C) = \mathrm{BCE}(\hat{C}, C) = \mathrm{BCE}\left(\frac{1}{255}\,\mathrm{BinaryDecod}(\hat{y}), C\right)$$

Methodology – Classification Loss



- Predicted probability $\hat{C}$ (or $\hat{y}$) of NFC does not strictly lie in (0, 1)

$$l_c(\hat{C}, C) = \mathrm{BCE}(\min(\hat{C}, 1-\epsilon), C)$$

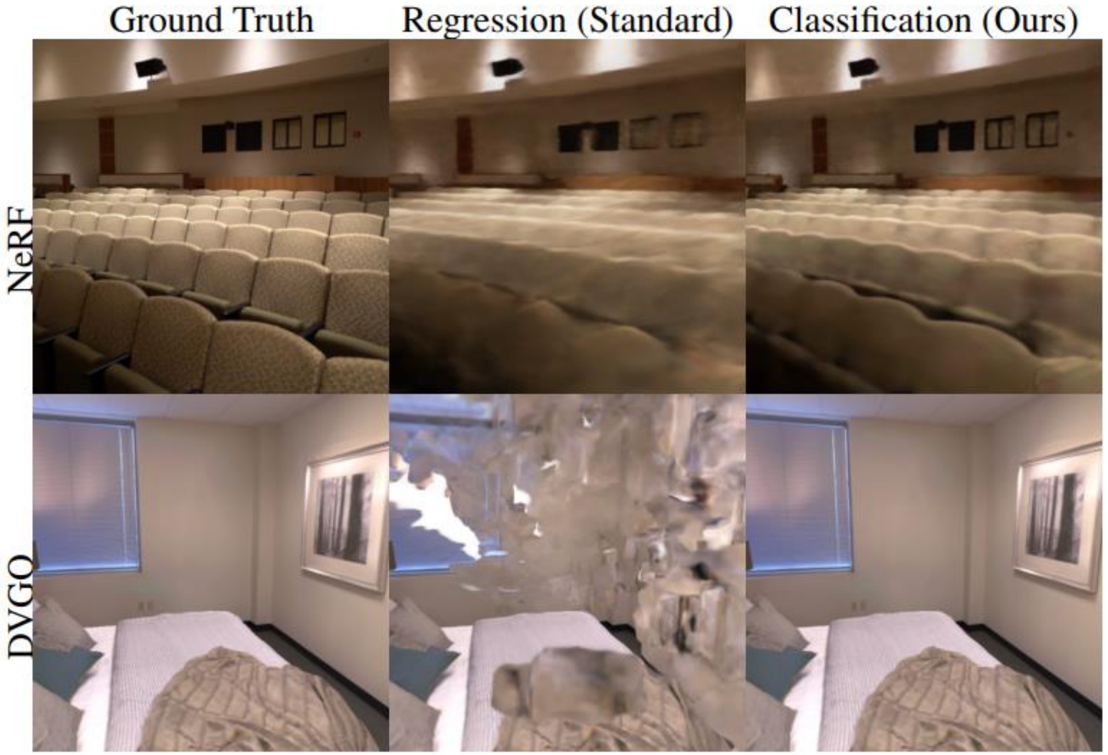
$$L_{\mathrm{NFC}}(\hat{C}, C) = \|\hat{C} - C\|_2^2 + \lambda\,\mathrm{BCE}(\min(\hat{C}, 1-\epsilon), C)$$

Experiments Settings

- Neural Field backbones
  - DVGO, vanilla NeRF, D-NeRF, NeuS
- Dataset
  - Static: Replica Dataset, Tanks and Temples Advanced
  - Dynamic: Lego and Hook
  - Challenging: Tanks and Temples Advanced
- Hyperparameters
  - Keep all hyperparameters same for them

Results

- Static Scene



Results

- Static Scene

Table 1: Quantitative results of DVGO on Replica Dataset.

Scene	Mode	PSNR(↑)	SSIM(↑)	LPIPS(↓)
Scene 1	Regression	13.03	0.508	0.726
	Classification	**34.63**	**0.934**	**0.0582**
Scene 2	Regression	14.81	0.654	0.640
	Classification	**33.82**	**0.942**	**0.0660**
Scene 3	Regression	15.66	0.661	0.634
	Classification	**34.04**	**0.965**	**0.0451**
Scene 4	Regression	18.17	0.696	0.546
	Classification	**36.52**	**0.977**	**0.0311**
Scene 5	Regression	15.17	0.640	0.504
	Classification	**35.93**	**0.974**	**0.0576**
Scene 6	Regression	21.33	0.854	0.254
	Classification	**29.75**	**0.941**	**0.0994**
Scene 7	Regression	22.54	0.865	0.231
	Classification	**34.77**	**0.966**	**0.0432**
Scene 8	Regression	15.89	0.724	0.519
	Classification	**33.40**	**0.952**	**0.0775**
Mean	Regression	17.08	0.700	0.507
	Classification	**34.11**	**0.956**	**0.0598**

Table 2: Quantitative results of DVGO, (vanilla) NeRF, NeuS on T&T Dataset. The mean PSNR, SSIM, LPIPS are computed over four scenes of T&T.

Model	Mode	PSNR(↑)	SSIM(↑)	LPIPS(↓)
DVGO	Regression	22.41	0.776	0.236
	Classification	**23.18**	**0.810**	**0.178**
NeRF	Regression	22.16	0.679	0.382
	Classification	**22.68**	**0.716**	**0.315**
NeuS	Regression	19.97	0.620	0.413
	Classification	**21.67**	**0.679**	**0.317**

Results

- Dynamic Scene
  - requires the capability of neural fields to model time domain.

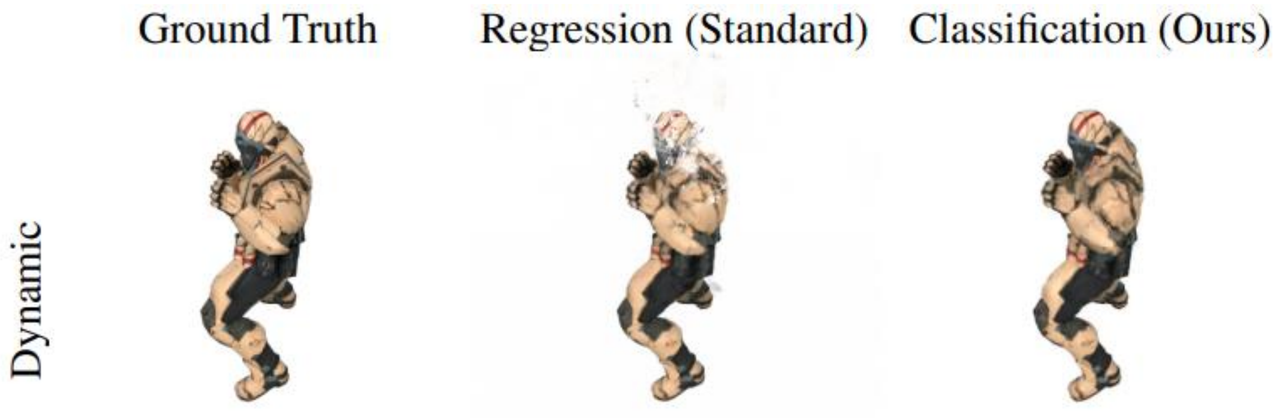


Table 3: Quantitative results of D-NeRF on dynamic scenes, LEGO and Hook.

Scene	Mode	PSNR(↑)	SSIM(↑)	LPIPS(↓)
Lego	Regression	21.64	0.839	0.165
	Classification	**23.11**	**0.886**	**0.121**
Hook	Regression	29.25	0.965	0.118
	Classification	**29.45**	**0.967**	**0.0392**

Results

- Challenging Scenes
  - Sparse: Fewer data for an object
  - Corrupted: Digital images often contain Gaussian noise

Table 4: Quantitative results of neural rendering with sparse training images.

Data Size	Mode	PSNR(↑)	SSIM(↑)	LPIPS(↓)
20%	Regression	14.87	0.530	0.580
	Classification	**19.38**	**0.629**	**0.395**
40%	Regression	18.76	0.637	0.426
	Classification	**21.73**	**0.711**	**0.340**
60%	Regression	21.02	0.682	0.394
	Classification	**22.27**	**0.728**	**0.329**
80%	Regression	21.72	0.698	0.386
	Classification	**22.46**	**0.734**	**0.322**

Results

- Challenging Scenes
  - Sparse: Fewer data for an object
  - Corrupted: Digital images often contain Gaussian noise

Table 5: Quantitative results of neural rendering with corrupted training images.

Noise Scale	Mode	PSNR(↑)	SSIM(↑)	LPIPS(↓)
0.2	Regression	21.97	0.694	0.406
	Classification	**22.33**	**0.719**	**0.353**
0.4	Regression	21.33	0.663	0.451
	Classification	**22.08**	**0.692**	**0.391**
0.6	Regression	19.67	0.615	0.512
	Classification	**21.45**	**0.662**	**0.429**

Ablation Study

$$L_{\mathrm{NFC}}(\hat{\boldsymbol{C}}, \boldsymbol{C}) = \|\hat{\boldsymbol{C}} - \boldsymbol{C}\|_2^2 + \lambda\, \mathrm{BCE}(\min(\hat{\boldsymbol{C}}, 1-\epsilon), \boldsymbol{C})$$

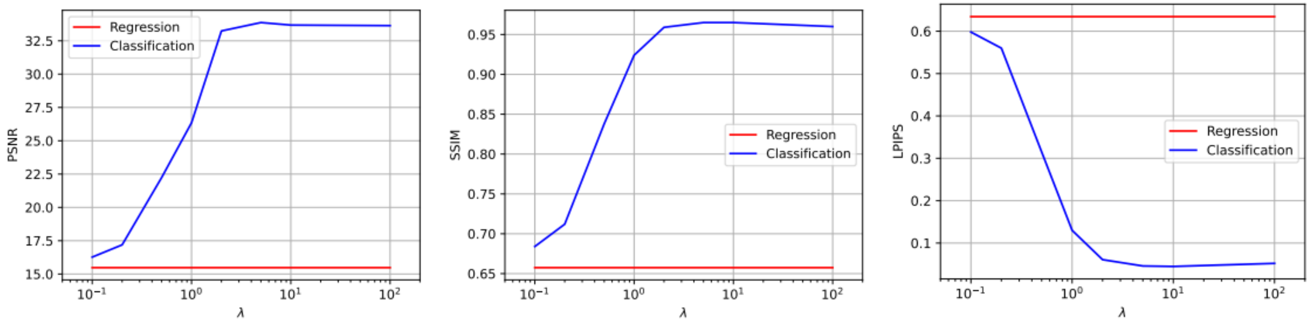


Figure 6: The curves of PSNR, SSIM, and LPIPS with respect to the hyperparameter λ. NFC is robust to a wide range of λ. Model: DVGO. Dataset: Replica Scene 3.

- Robustness to the hyperparameter
  - The quantitative results shows that a wide value range of λ can enhance the performance.

Limitation

- Target Encoding
  - In channel-wise classification loss
    - Is distance(relevant information) of binary-number system maintained?

$$l_c(\hat{\boldsymbol{C}}, \boldsymbol{C}) = \mathrm{BCE}(\hat{\boldsymbol{C}}, \boldsymbol{C}) = \mathrm{BCE}\left(\frac{1}{255}\,\mathrm{BinaryDecod}(\hat{\boldsymbol{y}}), \boldsymbol{C}\right)$$

$$\begin{array}{c} 1111 \\ \updownarrow \\ 0001 \\ \updownarrow \\ 1000 \end{array}$$

- Ignore the properties of the Neural Field.
  - Continuous!

Q & A