Enhancing Implicit Neural Representations via Symmetric Power Transformation

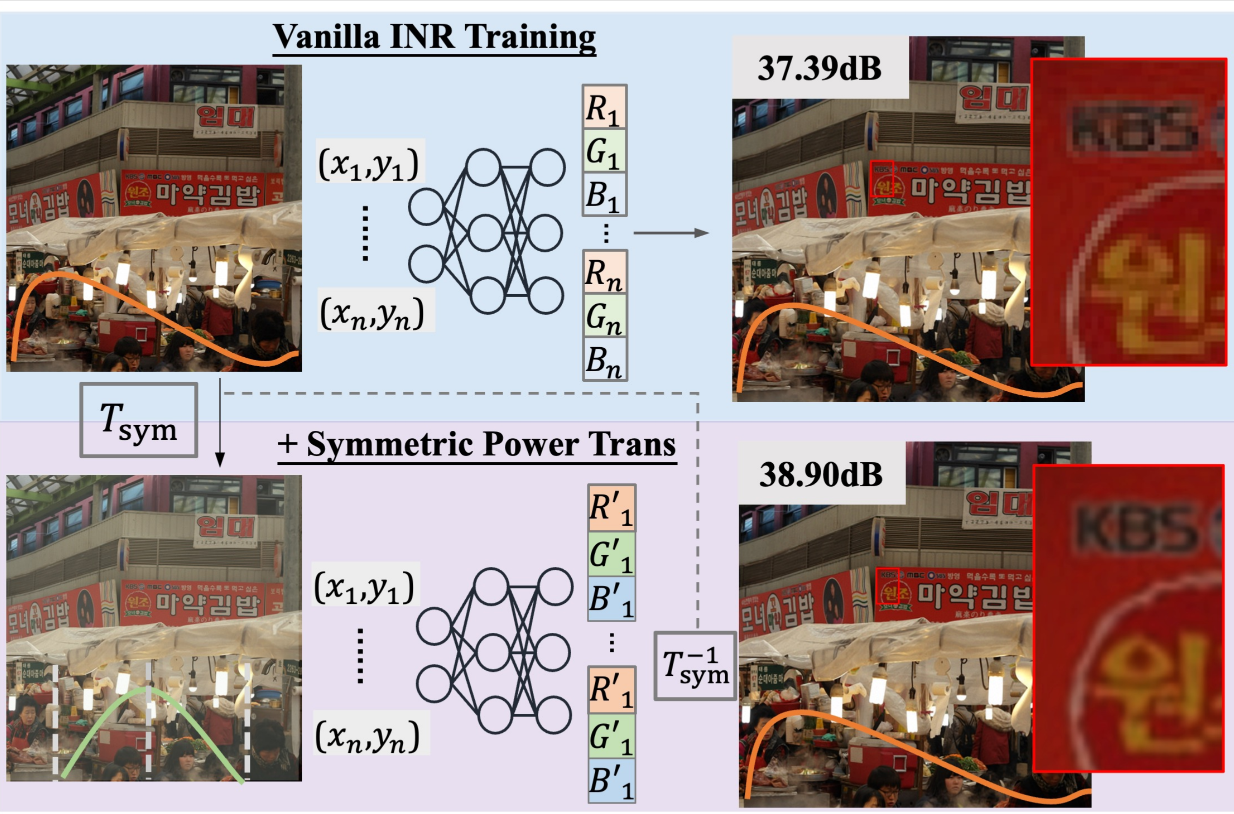
Weixiang Zhang, Shuzhao Xie, Chengwei Ren, Shijia Ge, Mingzi Wang, Zhi Wang

Background

Implicit Neural Representations (INRs) have been proposed for continuously representing signals using neural networks, which has garnered significant attention in the data representation.

Challenge:
Encoding signals into neural representations is resource-intensive, requiring training a neural network to fit natural signals.

Overview

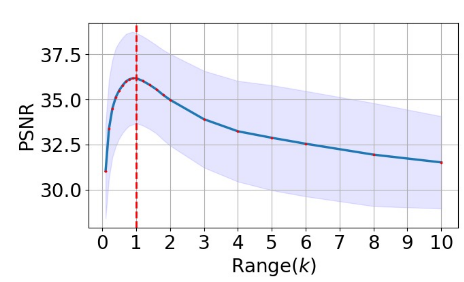


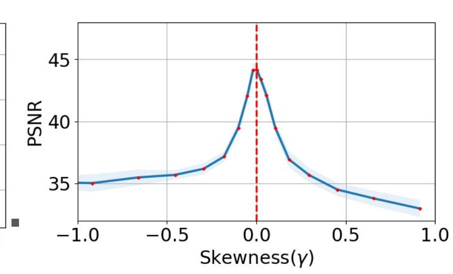
Contributions

- We observe that scaling the data to a specific **range** and ensuring a **symmetric** distribution benefits the training of INRs.
- We propose Symmetric Power Transformation to enhance implicit neural representation. We also introduce deviation-aware calibration and adaptive soft boundary to further improve the robustness of the method.
- We verify the effectiveness of our method through extensive experiments, including 1D audio fitting, 2D image fitting, and 3D video fitting task.

Range-Defined Symmetric Hypothesis

Given an INR F_θ with a bounded periodic activation function $\sigma(\cdot) \sim I$ and an input signal $\mathbf{y}$ with distribution $\boldsymbol{G}$, satisfying the following conditions can enhance the expressive ability of INRs: (1) **Range-Defined:** the bound of $\mathbf{y}$ is approximately I. (2) **Symmetric:** the skewness of $\boldsymbol{G}$ is approximately 0.





Symmetric Power Transformation

- **Basic Form:**

$$T_{\text{sym}}(\mathbf{y}) = (b-a)\mathbf{y}_0^{\beta} + a, \quad \mathbf{y}_0 = \frac{\mathbf{y} - \min(\mathbf{y})}{\max(\mathbf{y}) - \min(\mathbf{y})}$$

C.D.F.

$$\beta = \frac{\log \lambda}{\log(Q_\lambda)}, \quad F_Y(Q_\lambda) = P(Y \le Q_\lambda) = \lambda$$

COST-FREE Acceleration for INR's Training!

- **Deviation-Aware Calibration**:

$$\beta^{+} = \beta - \Delta\beta = \beta - \xi \int_0^{\tau} [f_{\text{sym}}(y) - f(y)]\mathrm{d}y$$

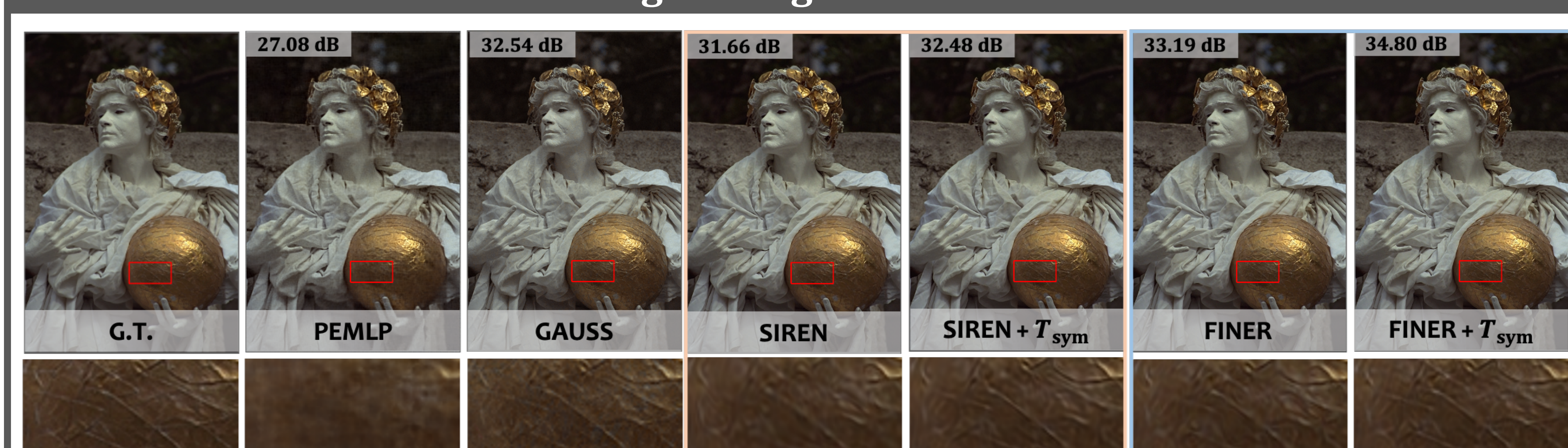
To calibrate extreme deviation boosting

- **Adaptive Soft Boundary:**

$$\mathbf{y}_0^{+} = \frac{\mathbf{y} - [1+\kappa f(0)]\min(\mathbf{y})}{[1+\kappa f(1)]\max(\mathbf{y}) - [1+\kappa f(0)]\min(\mathbf{y})}$$

To mitigate continuity-breaking at boundaries

Visualization for 2D natural image fitting



Please Visit:

arXiv

GitHub