SASNet: Spatially-Adaptive Sinusoidal Networks for INRs

Haoan Feng[1], Diana Aldana[2], Tiago Novello[2], Leila De Floriani[1]
[1]University of Maryland, College Park, [2] IMPA

hfengac@umd.edu, diana.aldana@impa.br,
tiago.novello@impa.br, deflo@umd.edu

Contents

1. Proof of Theorem 1

In the main paper, we claim that the j-th neuron of the $i+1$-th layer h_j^{i+1} admits the following expansion:

$$h_j^{i+1}(\mathbf{x}) = \sum_{\mathbf{k}\in\mathbb{Z}} \alpha \sin\left(\langle \mathbf{k}, \mathbf{y}\rangle + b_j^{i+1}\right) \tag{1}$$

where $\mathbf{y} = \mathbf{W}^i\mathbf{h}^i(\mathbf{x}) + \mathbf{b}^i$, and $\alpha = \prod_l J_{k_l}\left(W_{jl}^{i+1}\right)$ are the products of Bessel functions of the first kind at the j-th row of the hidden matrix $\mathbf{W}^{i+1}$.

Considering $h_j^{i+1} = \sin\left(\sum_{k=1}^m W_{jk}^{i+1}\sin(y_k) + b_j^{i+1}\right)$, we now prove Eq. (1) by induction on m. For $m = 1$,

$$\begin{aligned}
h_j^{i+1} &= \sin\left(W_{j1}^{i+1}\sin(y_1) + b_j^{i+1}\right)\\
&= \sin\left(W_{j1}^{i+1}\sin(y_1)\right)\cos(b_j^{i+1}) +\\
&\quad \cos\left(W_{j1}^{i+1}\sin(y_1)\right)\sin(b_j^{i+1})\\
&= \sum_{k\in\mathbb{Z}\,\text{odd}} J_k(W_{j1}^{i+1})\sin(ky_1)\cos(b_j^{i+1}) +\\
&\quad \sum_{k\in\mathbb{Z}\,\text{even}} J_k(W_{j1}^{i+1})\cos(ky_1)\sin(b_j^{i+1})\\
&= \sum_{k\in\mathbb{Z}\,\text{odd}} J_k(W_{j1}^{i+1})\sin(ky_1 + b_j^{i+1}) +\\
&\quad \sum_{k\in\mathbb{Z}\,\text{even}} J_k(W_{j1}^{i+1})\sin(ky_1 + b_j^{i+1})\\
&= \sum_{k\in\mathbb{Z}} \alpha\sin(ky_1 + b_j^{i+1}).
\end{aligned}$$

The first equality follows from the trigonometric identity $\sin(a+b) = \sin(a)\cos(b) + \cos(a)\sin(b)$, the second follows from standard identities for Bessel functions of the first kind (see [2, p. 361]). For the third, we use the properties $\sin(a)\cos(b) = \frac{\sin(a+b)+\sin(a-b)}{2}$ and $J_{-k}(a) = (-1)^k J_k(a)$ to rewrite the summations.
Now assume that Eq. (1) holds for $m-1$. Then,

$$\begin{aligned}
h_j^{i+1} &= \sin\left(\sum_{k=1}^m W_{jk}^{i+1}\sin(y_k) + b_j^{i+1}\right)\\
&= \sin\left(\sum_{k=1}^{m-1} W_{jk}^{i+1}\sin(y_k) + b_j^{i+1}\right)\cos\left(W_{jm}^{i+1}\sin(y_m)\right) +\\
&\quad \cos\left(\sum_{k=1}^{m-1} W_{jk}^{i+1}\sin(y_k) + b_j^{i+1}\right)\sin\left(W_{jm}^{i+1}\sin(y_m)\right)\\
&= \sum_{\substack{\mathbf{k}\in\mathbb{Z}^{m-1}\\ l\in\mathbb{Z}\,even}} \alpha_{\mathbf{k}} J_l(W_{jm}^{i+1})\sin\left(\sum_{s=1}^{m-1} k_s y_s + b_j^{i+1}\right)\cos(ly_m) +\\
&\quad \sum_{\substack{\mathbf{k}\in\mathbb{Z}^{m-1}\\ l\in\mathbb{Z}\,odd}} \alpha_{\mathbf{k}} J_l(W_{jm}^{i+1})\cos\left(\sum_{s=1}^{m-1} k_s y_s + b_j^{i+1}\right)\sin(ly_m)\\
&= \sum_{\mathbf{k}\in\mathbb{Z}^m} \alpha\sin(\langle \mathbf{k}, \mathbf{y}\rangle + b_j^{i+1})
\end{aligned}$$

Again, the second equality follows from the angle-sum identity for sine, and the third uses the induction hypothesis. This completes the proof.

2. Contribution map

Analyzing the contribution of each neuron is an effective approach to understand a network's fitting capability and parameter redundancy. According to Padmanabhan et al. [10], for a neuron h_j^i, its contribution to each pixel coordinate $\mathbf{x}$ is defined by a mapping from its weights $\mathbf{W}_j^i$ to the neuron output scalar $h_j^i(\mathbf{x})$, *i.e.*, $\sum_l h_j^i(\mathbf{x}) \cdot \mathbf{W}_{jl}^{i+1}$. Thus, the contribution map of h_j^i is given by accumulating the pixel-wise contribution across the entire image. These maps illustrate the spatial representation focus of neurons[1].

In the main paper, Fig. 3 presents contribution maps of nine neurons from SIREN [12], WIRE [11], and SASNet, representing different INR categories: *Global*, *Local*, and *Hybrid* (see more examples in the Related work). The maps reveal that neurons in *global* INRs tend to be redundant and *"space-agnostic"*, whereas those in *local* INRs exhibit sparser and more spatially localized contributions. Additionally, the global nature of INRs gives rise to model overfitting - neurons fitting high-frequency regions introduce noise into low-frequency areas. In such regions, *e.g.*, image backgrounds, other neurons may need to counteract the effect of high-frequency noise by learning complementary signals to suppress these artifacts. This inefficiency in INRs is one of the main issues addressed in this paper.

3. Implementation details

3.1. Frequency embedding layer initialization

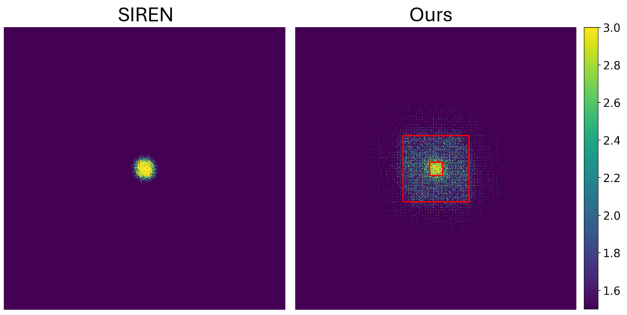

Figure 1. Comparing the frequency spectra of SIREN and our SASNet before training. Two red rectangles indicate band limits of low- and high-frequency embedding neurons.

As shown in Fig. 1, given a predefined band limit $\mathcal{B}$ and a low-frequency range $\mathcal{L}$, we split the frequency domain for selecting ω into two concentric rectangular regions giving rise to the low- and high-frequency neurons. Considering image-fitting as an example, low-frequency neuron multipliers $\mathbf{k}_{low} \in \mathbb{Z}^2$ are randomly sampled from $[-\mathcal{L}, \mathcal{L}]^2$. To ensure better coverage, high-frequency multipliers $\mathbf{k}_{high}$ are sampled on an integer grid within $[-\mathcal{B}, \mathcal{B}]^2$, excluding the low-frequency region $[-\mathcal{L}, \mathcal{L}]^2$. To reduce redundancy, we constrain sampling to positive integers along the x-axis, leveraging a symmetry of the frequency domain, where $(k_x, -k_y)$ is equivalent to $(-k_x, k_y)$. By adjusting $\mathcal{B}$ and balancing low- and high-frequency neurons, SASNet explicitly controls the output frequencies, improving high-frequency detail reconstruction. Choosing an appropriate $\mathcal{B}$ mitigates overfitting and enhances the reconstruction of high-frequency details.

3.2. Model configurations

For the image fitting task, we follow each method's open-source implementation unless stated otherwise. For the sinusoidal baselines, *i.e.*, FINER, SIREN, and SASNet, we increase ω_0 from 30 to 43 because the images used here are of higher resolution than those in prior work, and this value gives a good trade-off between PSNR and reconstruction noisiness. All models are optimized with Adam using a fixed learning rate of 10^{-4}, except as noted below.

Table 1 summarizes the task-specific SASNet hyperparameters used in our image fitting, volumetric fitting, and SDF reconstruction experiments. In all cases, the low-frequency mask is fixed to one, the remaining masks are initialized around 0.5, and the hash-grid and decoder use a learning rate of 5×10^{-4}.

Table 1. SASNet model configurations used in the supplementary experiments. For the SDF task, the band limit is 30 for Armadillo and Dragon, and 40 for Lucy and Thai Statue.

Setting	Image	Volume	SDF
Hardware	RTX A6000	RTX A6000	RTX A6000
Batch size	262144	1780000	10000
Input dim	2	3	3
ω_0	43	30	30
# trainable params	128865	138181	130917
FEmb layer size	256	256	256
Hidden layer size	232	192	224
Band limit	60	30	30 (40)
Low-freq range	12	6	6
Low-freq percentage	0.5	0.5	0.5
# high-freq bands	4	4	4
# neuron groups	8	8	8
Hash-grid			
feature dim	2	2	2
feature std	0.01	0.01	0.01
feature bias	0	0	0
codebook bitwidth	10	11	10
num levels	10	12	10
Mask decoder	ReLU-MLP with one hidden layer of size 48		

Additional implementation details for the compared methods are listed below:

- FINER [8]: the bias scale is set to 0.
- FFN [13] and SAPE [5]: we use the implementation in SAPE [5][2]. For each training step, instead of sampling coordinates, we feed all coordinates for supervision. The number of frequencies is set to 256 according to its performance on the Kodak dataset.
- WIRE [11]: we use the implementation that each network layer uses a complex Gabor function (except the output layer). The learning rate is fixed to 5e-4. For the image-fitting task, the hyperparameter `omega` is set to 20, and `sigma` is set to 30, as suggested by its implementation. For the toy example, the

[1] Publicly available at https://github.com/namithap10/xinc

[2] Publicly available at https://github.com/amirhertz/SAPE.

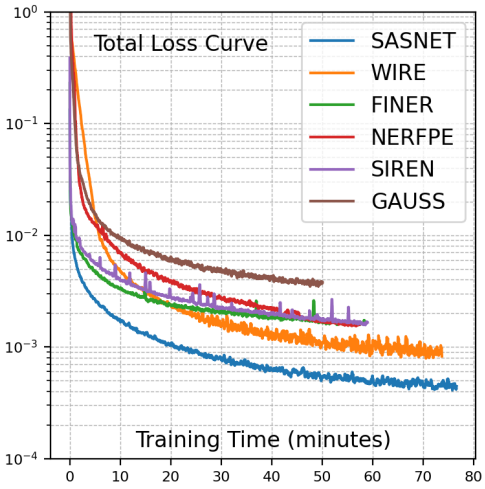

Figure 2. Training loss plotted against wall-clock time on the 2D image fitting task. SASNet reaches the same loss level substantially faster, highlighting its improved time-to-accuracy despite a modest per-step overhead.

hyperparameter omega is set to 4, and the sigma is set to 10 according to a simple hyperparameter search.

- GaussianImage [14]: without applying quantization, we fitted each image with 16640 Gaussians for 50K training epochs.
- NeuRBF [4]: we tune the hyperparameters and set `log2_hashmap_size` to 14 and `n_levels` to 10.
- SASNet: the full task-specific config. is summarized in Tab. 1.

For volumetric data and SDF fitting, we use the implementation of FINER [8][3] and reuse its training and evaluation scripts on the four provided training point clouds. The corresponding SASNet hyperparameters for these two tasks are listed in Tab. 1.

4. Computation performance and overhead

Table 2. Efficiency comparison on the 2D image fitting task. We report average step time (AST) and peak GPU memory usage (PMem) for models with comparable numbers of trainable parameters. Lower values indicate better efficiency.

Model	AST↓	PMem↓
SIREN	**29,224**	**2075.50**
MSIREN	95,438	6077.07
INCODE	59,907	3642.59
SASNET	43,107	3031.60

We report average step time (AST,*us*) and peak GPU mem usage (PMem, `MiB`) from PyTorch Profiler (averaged over 5 runs) with similar #trainable params. For the image fitting task (Tab. 2), we use batch size 262,144; INCODE[7] freezes the `feature_extractor` and MSIREN[9] assumes precomputed latent codes and uses 32^2 tiles with stride 16. For the 3D shape fitting task, we further profile SASNet and SIREN. SASNet increases AST from 17,557 to 22,175 and PMem from 98.0 to 118.2, indicating only a modest computational overhead.

This overhead mainly comes from the multi-scale hash-grid queries used to generate spatial masks. In practice, these gathers are often latency-limited because their memory accesses exhibit poor spatial locality and are therefore not well coalesced, making runtime sensitive to cache misses and memory-level parallelism. However, per-step efficiency alone does not fully reflect optimization behavior. When we redraw the training curve in the main figure as Fig. 2 with wall-clock time, SASNet reaches the same training loss in approximately 10 minutes, whereas SIREN requires 58 minutes. Overall, SASNet offers a favorable trade-off between model complexity and training efficiency: it incurs a moderate increase in runtime and memory usage, but converges substantially faster in practice. A more detailed runtime breakdown and optimization study is left for future work. In addition, the signal-conditioned modulation strategy used in INCODE suggests a possible direction for reducing this overhead further, namely predicting spatial masks with an INCODE-style module instead of generating them explicitly.

Table 3. Quantitative evaluation on the Kodak [1] dataset.

Methods	PSNR ↑	SSIM ↑	$\text{PSNR}_{\text{edge}}$ ↑	Noisiness ↓
SIREN [12]	36.60	0.951	33.28	4.079
FFN [13]	33.00	0.868	29.87	10.353
SAPE [5]	33.37	0.872	31.65	5.373
WIRE [11]	29.34	0.804	25.23	15.375
FINER [8]	37.72	0.973	34.67	5.174
GI [14]	37.58	0.971	32.59	-
NeuRBF [4]	37.51	0.971	34.78	6.258
SASNet	40.43	0.986	38.12	3.993

5. More experimental results

5.1. Influence of the hyperparameter ω_0

As shown in Fig. 1 of the main paper, SIREN [12] is highly sensitive to the choice of the hyperparameter ω_0, making its selection non-trivial. Smaller values lead to blurry reconstructions without introducing high-frequency noise, whereas higher values reduce spectral bias and produce sharper outputs, but resulting in increased noisiness in smooth regions.

In contrast, SASNet leverages spatially-adaptive masks to separate frequency band expressions, preventing high-frequency neuron outputs from propagating into smooth regions. As a result, the choice of ω_0 has minimal impact on the final reconstruction accuracy, as illustrated in Fig. 3, where PSNR remains stable across different ω_0 values.

5.2. Visualizations for the SDF

Figure 4 first visualizes the SDF error map for the Armadillo example. We compare a model equipped with the fixed frequency embedding but without spatial masks against the full SASNet model. Without spatial localization, the reconstruction error is

[3]Publicly available at: https://github.com/liuzhen0212/FINER

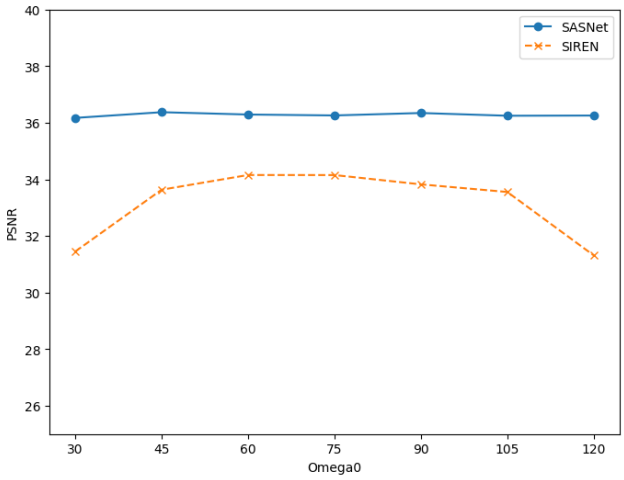

Figure 3. Sensitivity comparison between SIREN and our SASNet to the change of hyperparameter ω_0.

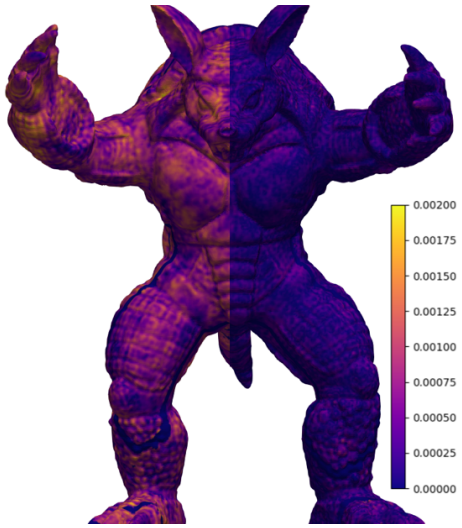

Figure 4. SDF error map for the Armadillo example. Left: a model with fixed frequency embedding but without spatial masks. Right: SASNet. The error map shows that spatially-adaptive masking suppresses widespread surface artifacts and localizes the remaining error to a smaller set of challenging regions.

distributed more broadly over the surface, especially around geometrically detailed regions, which indicates that globally available high-frequency components are not used selectively. In contrast, SASNet suppresses these artifacts and produces a substantially cleaner error distribution, showing that the learned masks help restrict high-frequency responses to the regions where they are most needed.

To further explain this behavior by the Dragon example in the main paper, Figure 5 visualizes the learned spatial mask corresponding to the *highest-frequency band* of the frequency embedding layer. The figure shows the target SDF isosurface for Dragon overlaid on the volumetric mask produced by the multi-scale hash-grid decoder. The mask values are rendered as a volumetric field using the same color encoding adopted for the 2D image masks. High mask responses concentrate near sharp geometric ridges, thin structures, and high-curvature surface regions, while remaining weak in smooth or empty space. This qualitative result confirms that SASNet allocates high-frequency capacity selectively, which helps explain the reduced error observed in Fig. 4.

5.3. Additional experiments on CT medical images

In addition to the two RGB image datasets used in the main paper, we further compare SASNet with FINER [8] and SIREN [12] on a publicly available CT medical image dataset[4], which contains 100 CT scan images from the TCIA cancer imaging archive [3]. Each scan has a resolution of 512^2 and consists of a single-channel grayscale image, normalized to $[0, 1]$. We conduct the image-fitting experiments using the same configuration as in the main paper. These results further demonstrate that SASNet generalizes well across modalities and resolutions.

Table 4 presents the quantitative evaluations, showing that SASNet achieves higher fitting accuracy, particularly in preserving edge details. Notably, for single-channel images, the model configuration used in previous experiments is sufficiently large for all methods to fit the data without overfitting. Consequently, the noisiness across all methods remains low. This is further observable in Fig. 6, where both the model outputs and their error maps are visualized.

Table 4. Quantitative evaluation on a medical image dataset. For each metric, ↑/↓ indicates the numerical value is *higher/lower* the method performs better.

Methods	PSNR ↑	SSIM ↑	$\text{PSNR}_{\text{edge}}$ ↑	Noisiness ↓
SIREN [12]	47.07	0.98	42.94	3.482
FINER [8]	48.49	0.99	45.56	3.488
SASNet (Ours)	50.21	0.99	49.58	3.681

5.4. Additional experiments on FFHQ dataset

The Flickr-Faces-HQ (FFHQ) dataset [6] contains a large collection of high-quality human face images at a resolution of 1024^2. For this experiment, 100 images are randomly selected from the training image set. Similar to the medical image experiment, the same network configuration is applied to SIREN [12], FINER [8], and our SASNet.

Table 5 presents the quantitative evaluations, while Fig. 7 provides a visual comparison among these methods. SASNet achieves higher fitting accuracy (as observed in the error maps), and this PSNR improvement does not result in an increased noisiness level, in contrast to FINER.

5.5. Additional ablation study results

In the main paper, we conduct an ablation study to evaluate the contribution of each module, namely the frequency embedding layer and spatially-adaptive masks at different layers. Figure 8

[4] https://www.kaggle.com/datasets/kmader/siim-medical-images

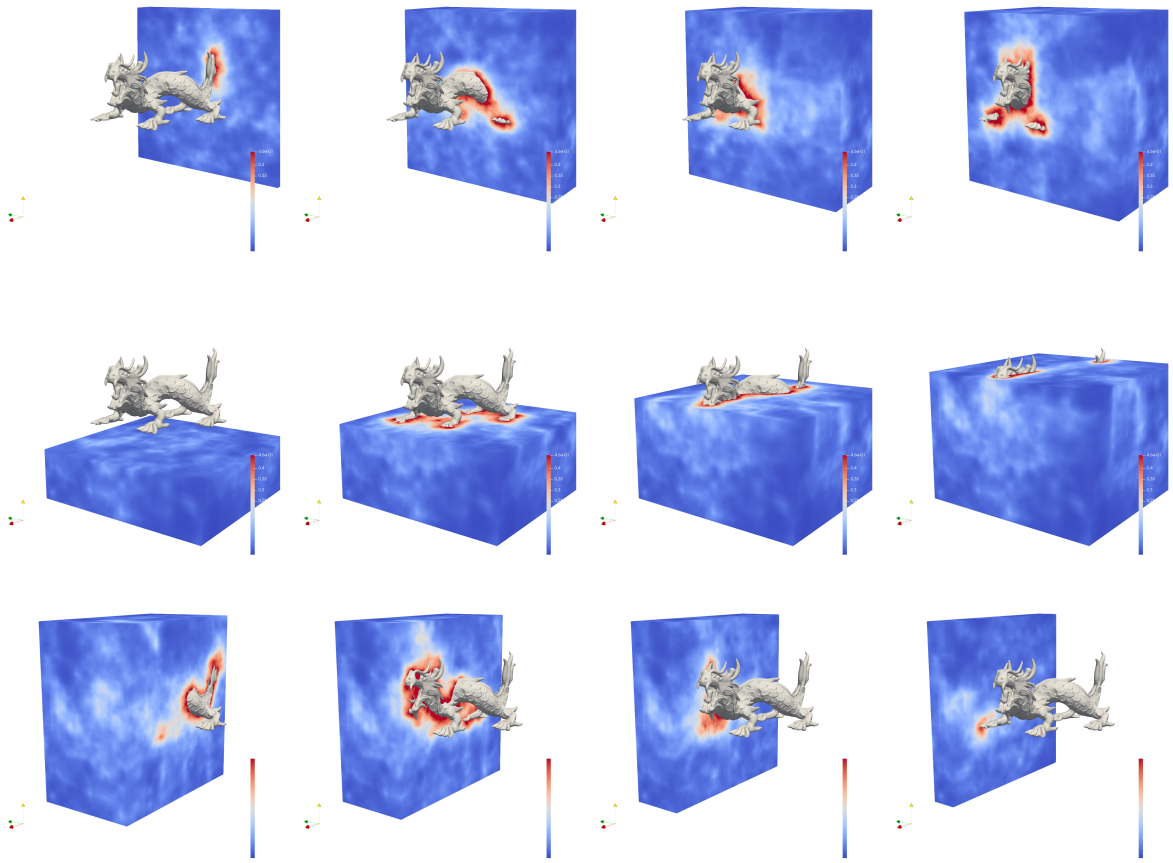

Figure 5. Learned 3D spatial mask for SASNet's highest-frequency band on the Dragon SDF. Across multiple viewpoints, high responses are concentrated near the object surface and around geometrically complex regions.

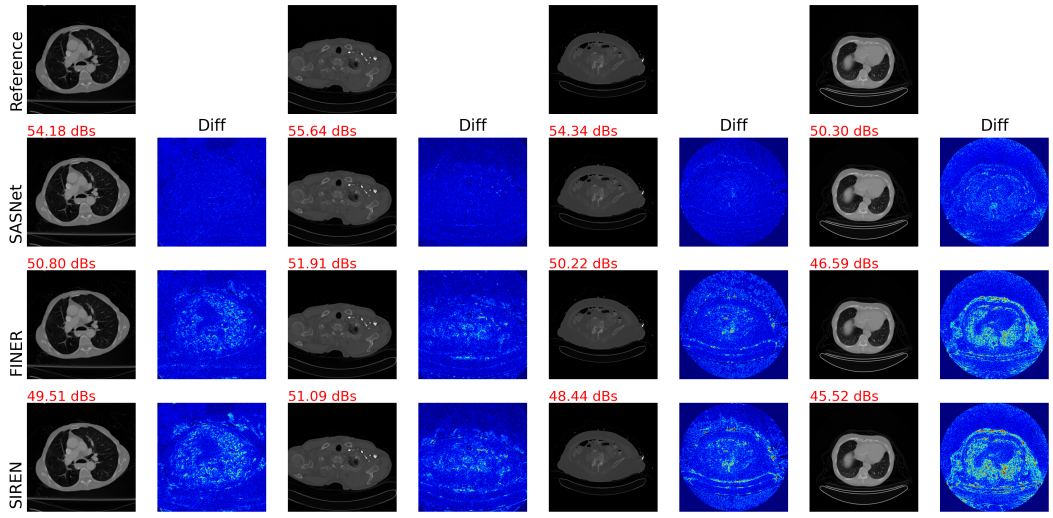

Figure 6. Medical CT scans' fitting results by SASNet, FINER, and SIREN. For each of the four examples, PSNR values are annotated on the upper left corner of the model outputs. Besides each model's output image, the *Diff* column shows the error map with a *jet* color encoding (up to 2% absolute difference).

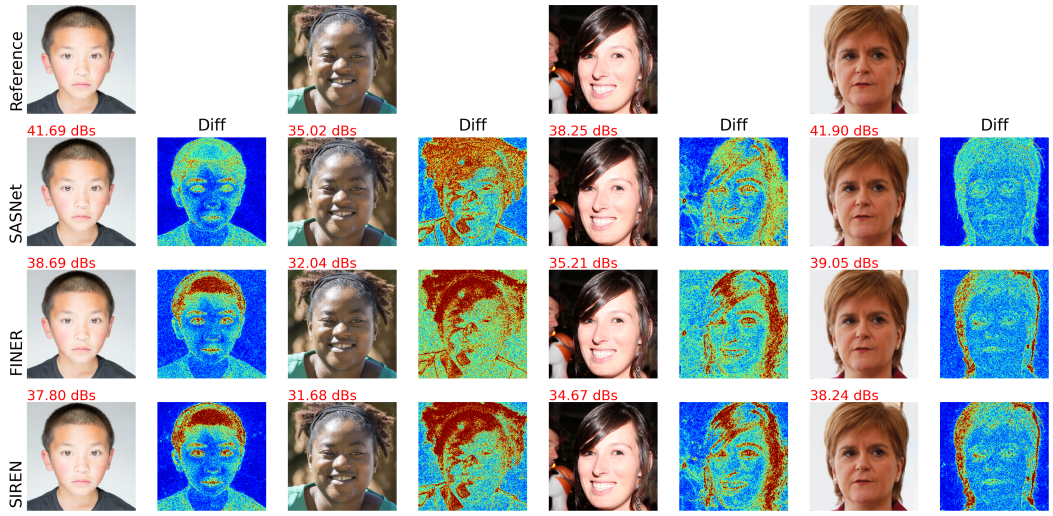

Figure 7. Fitting results of FFHQ human faces by SASNet, FINER, and SIREN. For each of the four examples, PSNR values are annotated on the upper left corner of model outputs. Besides each model's output image, the *Diff* column shows the error map with a *jet* color encoding (up to 2% absolute difference).

Figure 8. Visual comparison between SIREN, SASNet without masks, and full SASNet. First row shows the image reconstructed and the second row shows the corresponding gradient intensity obtained by network automatic differentiation (*autograd* in PyTorch).

presents a visual comparison among SIREN, SASNet with only frequency embedding (without masks), and the full SASNet. The results demonstrate that while the frequency embedding layer enhances fitting accuracy, particularly for sharper reconstructions, it also introduces additional noise in the image background. By applying spatially-adaptive masks, the quality of background reconstruction significantly improves, as indicated by the reduction in image gradient intensity.

Figure 9 visualizes the jointly optimized spatial masks of SASNet and SAPE for the same input image as in Fig. 8. The learned

Table 5. Quantitative evaluation on the FFHQ dataset. For each metric, ↑/↓ indicates the numerical value is *higher/lower* the method performs better.

Methods	PSNR ↑	SSIM ↑	$\text{PSNR}_{\text{edge}}$ ↑	Noisiness ↓
SIREN [12]	37.58	0.95	34.20	4.178
FINER [8]	38.16	0.95	35.09	4.975
SASNet (Ours)	39.69	0.95	37.64	4.278

masks for different frequency bands effectively align with the spatial complexity of the image, enabling the model to focus its fitting capacity on complex, high-frequency regions while preventing resource waste and overfitting in simpler regions.

Additionally, a hidden mask is selected from each hidden layer, and contribution maps of the corresponding neuron groups are visualized. These results indicate that spatial masks allow different regions to encode distinct frequency bands, leading to lower network redundancy in SASNet compared to SIREN. In smooth and easy-to-fit regions, only a subset of neurons contributes to reconstruction, whereas in complex, texture-rich regions, most neurons participate in signal reconstruction, demonstrating the effectiveness of spatially-adaptive frequency control.

6. Extreme cases

Under extreme cases, as shown in Fig. 10, INRs' performance can be limited for certain classes of image content. Both images primarily consist of high-frequency repeated patterns that span almost the entire domain, and with minimal pixel variation within each "color strip". This structure is particularly well-suited for GaussianImage, as each Gaussian kernel can efficiently cover a larger portion of the domain with minimal computational effort. Conversely, INR methods suffer from spectral bias and an insufficient number of parameters to effectively capture such a broad frequency spectrum, leading to reduced fitting accuracy in these scenarios. These examples illustrate the inherent limitations of INRs under extreme global-frequency conditions.

References

[1] Kodak lossless true color image suite. http://https://r0k.us/graphics/kodak/, 1999. Accessed: 2024-09-01. 3

[2] Milton Abramowitz and Irene A Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, volume 55. US Government printing office, 1948. 1

[3] Brad Albertina, Mark Watson, Chandra Holback, Rose Jarosz, Shanah Kirk, Yueh Lee, Kimberly Rieger-Christ, and John Lemmerman. The cancer genome atlas lung adenocarcinoma collection (tcga-luad). 2016. 4

[4] Zhang Chen, Zhong Li, Liangchen Song, Lele Chen, Jingyi Yu, Junsong Yuan, and Yi Xu. Neurbf: A neural fields representation with adaptive radial basis functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4182–4194, 2023. 3

[5] Amir Hertz, Or Perel, Raja Giryes, Olga Sorkine-Hornung, and Daniel Cohen-Or. Sape: Spatially-adaptive progressive encoding for neural optimization. *Advances in Neural Information Processing Systems*, 34:8820–8832, 2021. 2, 3

[6] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 4

[7] Amirhossein Kazerouni, Reza Azad, Alireza Hosseini, Dorit Merhof, and Ulas Bagci. Incode: Implicit neural conditioning with prior knowledge embeddings. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1298–1307, 2024. 3

[8] Zhen Liu, Hao Zhu, Qi Zhang, Jingde Fu, Weibing Deng, Zhan Ma, Yanwen Guo, and Xun Cao. Finer: Flexible spectral-bias tuning in implicit neural representation by variable-periodic activation functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2713–2722, 2024. 2, 3, 4, 7

[9] Ishit Mehta, Michaël Gharbi, Connelly Barnes, Eli Shechtman, Ravi Ramamoorthi, and Manmohan Chandraker. Modulated periodic activations for generalizable local functional representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14214–14223, 2021. 3

[10] Namitha Padmanabhan, Matthew Gwilliam, Pulkit Kumar, Shishira R Maiya, Max Ehrlich, and Abhinav Shrivastava. Explaining the implicit neural canvas: Connecting pixels to neurons by tracing their contributions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10957–10967, 2024. 2

[11] Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G Baraniuk. Wire: Wavelet implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18507–18516, 2023. 2, 3

[12] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020. 2, 3, 4, 7

[13] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020. 2, 3

[14] Xinjie Zhang, Xingtong Ge, Tongda Xu, Dailan He, Yan Wang, Hongwei Qin, Guo Lu, Jing Geng, and Jun Zhang. Gaussianimage: 1000 fps image representation and compression by 2d gaussian splatting. In *European Conference on Computer Vision*, 2024. 3

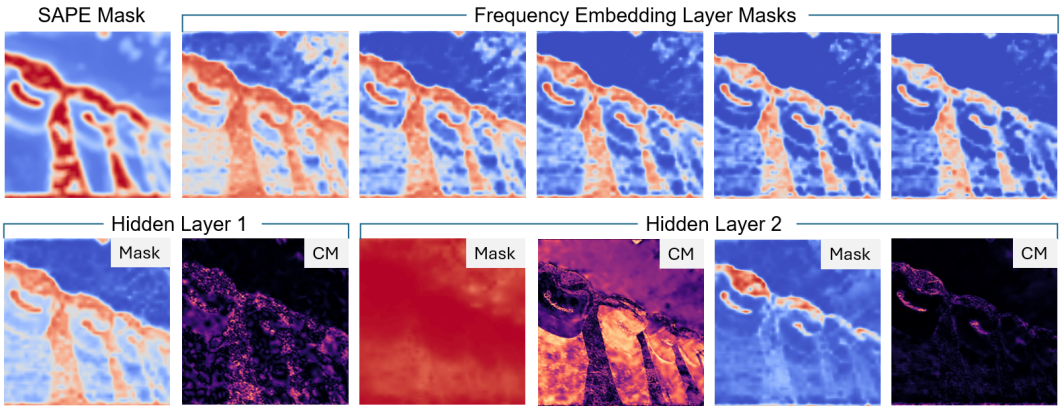

Figure 9. SAPE mask and our masks of the image-fitting task. **CM**: contribution map. The first hidden layer mask shows that this group has a global contribution, as also shown in the CM. However, some details on edges and texture regions can be missed, and therefore, the other group focuses only on these small regions.

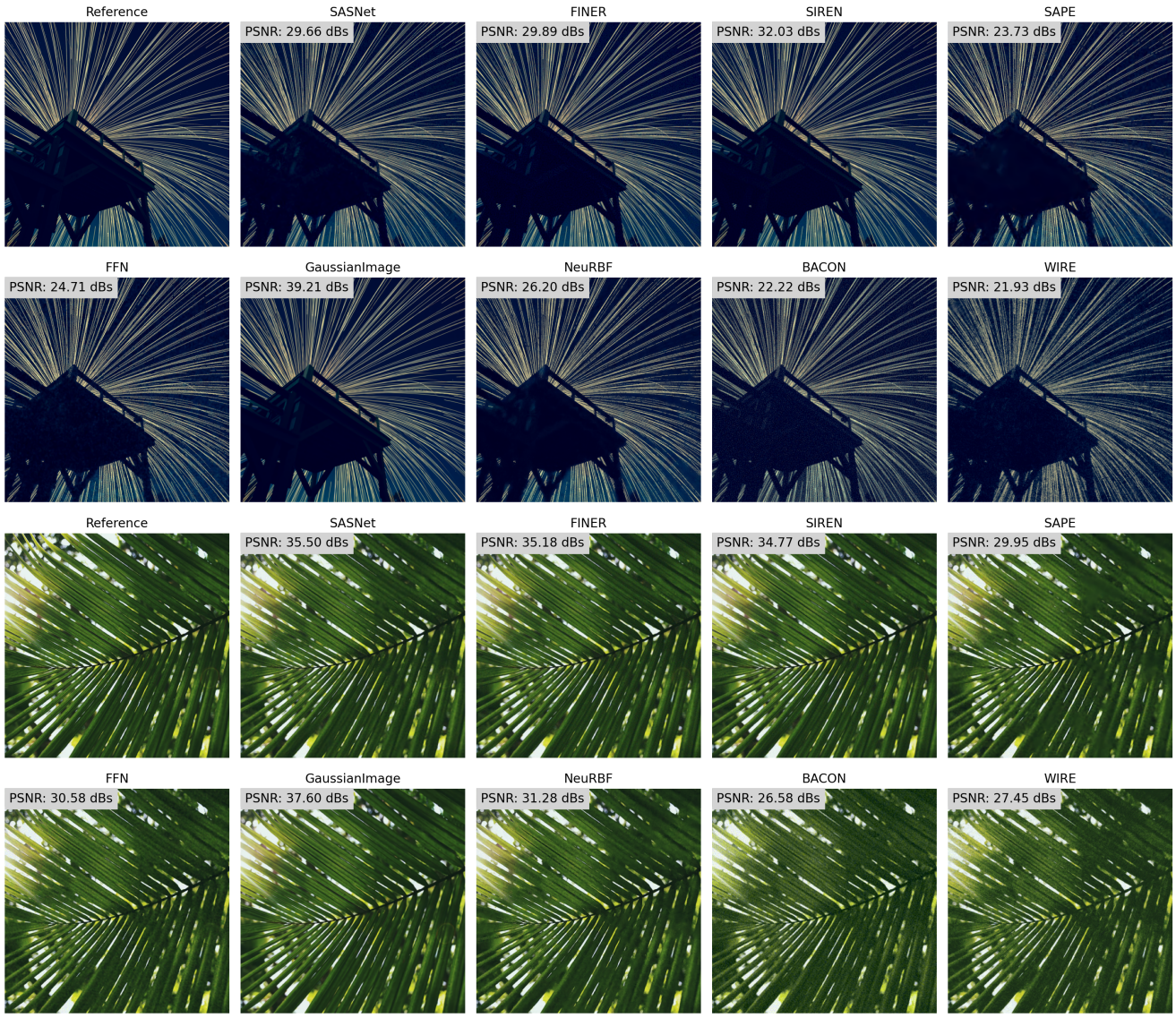

Figure 10. Example images from the DIV2K dataset showing when performances of INRs are limited. Both images are of complex signal components and higher frequencies than others, and contain hard edges covering almost the whole domain.