

Filtering In Implicit Neural Networks

Yixin Zhuang

Abstract

Implicit neural networks (INNs) are very effective for learning data representation. However, most INNs inevitably generate over-smoothed patches or obvious noisy artifacts in the results when the data has many scales of details or a wide range of frequencies, leading to significant performance reduction. Adapting the result containing both noise and over-smoothed regions usually suffers from either over smoothing or noisy issues. To overcome this challenge, we propose a new framework, coined FINN, that integrated a *filtering* module to the *implicit neural network* to perform data fitting while filtering artifacts. The filtering module has a smoothing operator that acts on the intermediate results of the network and a recovering operator that brings distinct details from the input back to the regions overly smoothed. The proposed method significantly alleviates over smoothing or noisy issues. We demonstrate the advantage of the FINN on the image regression task, considering both real and synthetic images, and showcases significant improvement on both quantitative and qualitative results compared to state-of-the-art methods. Moreover, FINN yields better performance in both convergence speed and network stability. Source code is available at <https://github.com/yixin26/FINN>.

1. Introduction

Filtering techniques are widely applied to many applications to remove the artifacts from the data, such as images and 3D shapes. For the data with many scales of details or a wide range of frequencies, it is challenging to recover the data while maintaining free of unwanted content using neural networks, therefore, filters are usually necessary for post-processing, or in the other way, additional strategies, e.g., coarse to fine generation, hierarchical representations, or scale-aware operations, are needed to adapt different scales of details. This paper addresses built-in learnable filters in an implicit neural network that achieves reconstruction and filtering purposes within a simple network framework.

An Implicit Neural Network (INN) (Mescheder et al., 2019;

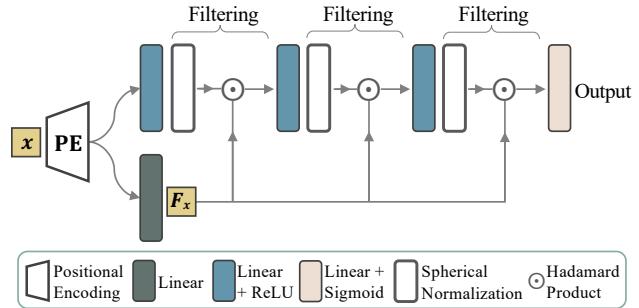


Figure 1. We present a new framework for learning implicit neural representation with filters. Filtering artifacts is usually at the expense of losing details. Aiming to achieve spatial adaptive filtering, we incorporate filtering modules into multilayer perceptrons (MLPs) to perform smoothing and recovering operators iteratively at each step of MLPs. The smoothing operator outputs over-smoothed results, while the recovering operator restores fine details. The combination of MLPs and filters consequently yield adaptive reconstructing of the target function with a wide range of frequency bandwidths.

Park et al., 2019; Chen & Zhang, 2019) parameterizes the data as a continuous function that maps spatial coordinates to the corresponding values under varying representations (e.g., RGB, signed distance, etc.). Such representation has infinite resolutions and is learned using simple network structures, i.e., multilayer perceptrons (MLPs) with ReLU activations. Moreover, INNs can fit functions with wide frequency bandwidths by encoding the coordinates to high-dimensional vectors, i.e., random Fourier features (FFN) (Tancik et al., 2020), or replacing ReLU activation functions with periodic functions, i.e., sinusoidal activation (SIREN) (Sitzmann et al., 2020), leading to significant improvements.

Despite the impressive progress of the recent INNs (e.g., FFN, SIREN), they can still have difficulties in recovering complex functions, generating either over-smoothed regions or noise in the results. Unfortunately, there is no ubiquitously valid filter to adapt the reconstruction to avoid these issues, and each data needs to be individually adjusted. To this end, some explicit spatial/local adapting schemes are contributed to INNs. For example, SAPE (Hertz et al., 2021) progressively encodes input with increasing frequencies for individual spatial coordinates to avoid using excessive fre-

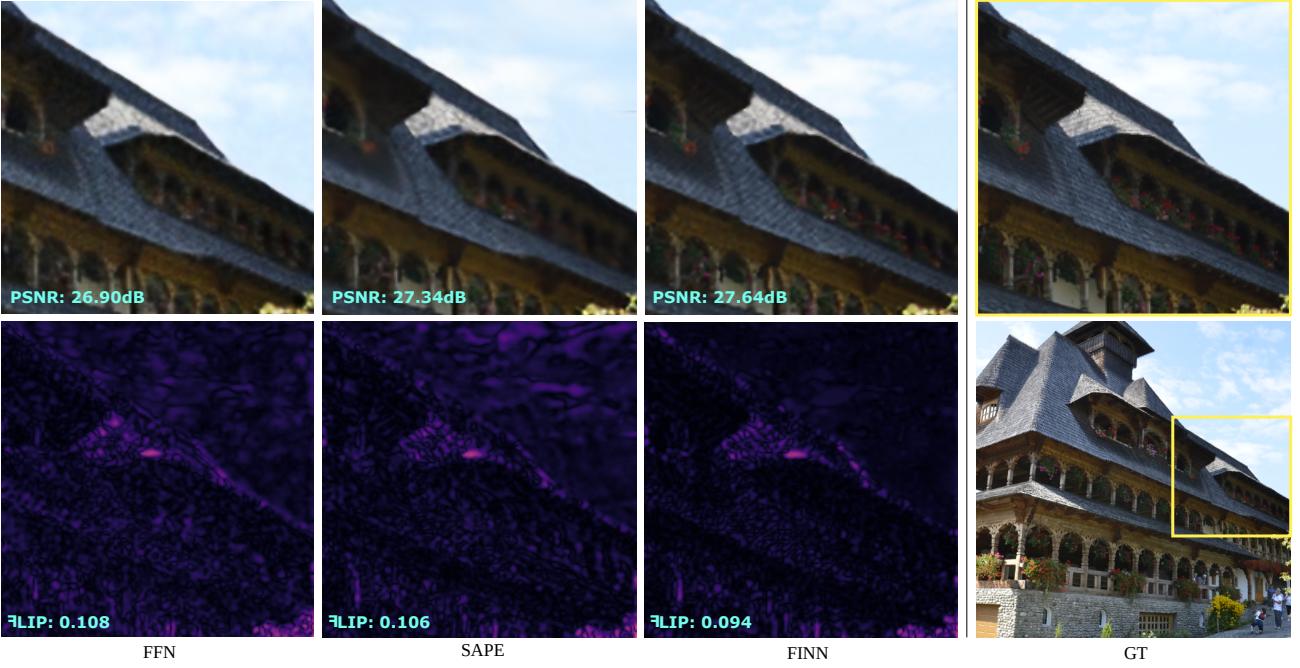


Figure 2. An image generated by Fourier Features Network (FFN (Tancik et al., 2020)) contains obvious noises (left top), which is better visualized by the error map computed by FLIP (Andersson et al., 2020) (left bottom). (Second column) spatial adaptation of frequency bands from SAPE smoothes out noise, but at the expense of losing wanted details, e.g., small clusters of clouds. To circumvent over-smoothing issue, our method (FINN) incorporates a filtering module that effectively removes artifacts while recovering distinct details (Third column). PSNR (higher is better) and FLIP (lower is better) measures the similarity between generated results and ground truth images.

quencies at smooth regions, and the work (Mehta et al., 2021) represents the input as a structured grid from which the coordinates are encoded conditioning on the corresponding grids. An example is shown in Figure 2, where explicit spatial adaptation method (i.e., SAPE) provides results with fewer noise compared to the baseline method (i.e., FFN). Despite the improvement, SAPE tends to over-smooth the result (e.g., the sky and the regions under roofs). Alternatively, this paper seeks implicit filtering functions to achieve spatial adaptation without discretized/subdivided coordinate space and circumvent overly smoothing issues observed in the explicit adaptation methods.

We define the filtering module as two consecutive operators, a smoothing operator, i.e., spherical normalization, overly smoothes the results, and a recovering operator, i.e., Hadamard product, brings fine details from the input back to the results, as shown in Figure 1 (FINN). The two operators together with the MLPs form a simple network function. Intuitively, spherical normalization flattens both global and small-scale variations, and the recovering operation enables small structures to be brought back from an initial domain to the smoothed results. With the filter, the results from MLPs are layer-wisely adjusted. Our method showcases improvement at regions with varying frequency bandwidths

compared to FFN and SAPE (Figure 2).

To demonstrate the effectiveness of the filtering module, we evaluate our model on the image regression task using both real and synthetic image datasets and showcase that our method can capture a wide range of frequencies better than state-of-the-art methods, including FFN, SAPE, and SIREN, and achieve significant performance improvement. Moreover, filtering in implicit neural networks yields a faster convergence speed and more stable performance on both network training and testing.

In this work, we make the following contributions.

- We propose a simple framework that extends MLP-based implicit neural network with filtering functions that performs filtering while reconstructing data.
- We devise a simple filtering function with iterative smoothing and recovering operations to each layer of MLPs to adjust the reconstruction at individual regions with different frequency bandwidths.

In addition, the whole network has a simple formulation that can be easily modified to realize different filtering effects (e.g. using an alternative recovering function).

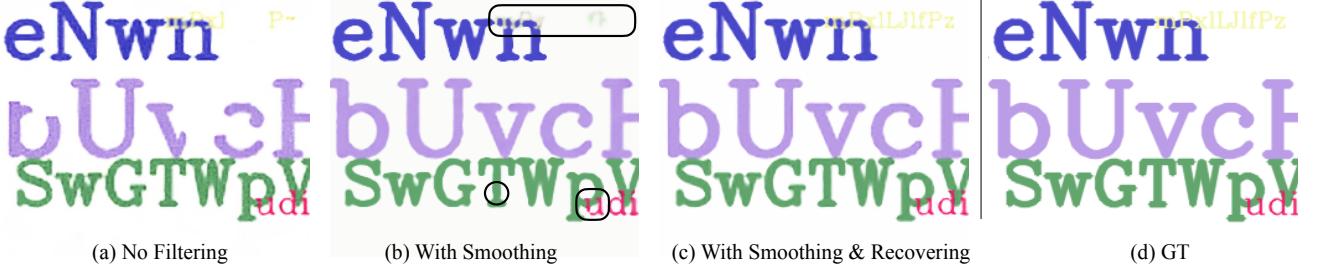


Figure 3. Reconstruction using different combinations of filtering operations (a-c). (a) is generated by FFN (Tancik et al., 2020) with no filtering. With smoothing operation (b), our method generates much complete and smoother texts than FFN. By applying both smoothing and recovering operations (c), small structures are recovered, e.g., the missing details highlighted in black circles in (b). Under the same sampling rate (25% for training), incorporating a filtering module significantly improves the visual quality of reconstructed results.

2. Related Work

Implicit Neural Representation (INR). Deep neural network has been shown to be highly effective for learning implicit function that represents images (Tancik et al., 2020; Sitzmann et al., 2020; Mildenhall et al., 2022) and 3D objects and scenes (Park et al., 2019; Mescheder et al., 2019; Chen & Zhang, 2019; Atzmon & Lipman, 2020; Sitzmann et al., 2019). They incorporate coordinates as inputs to Multi-Layer Perceptrons (MLPs) that could be sampled to an arbitrarily high spatial resolution. Thus INR can directly be applied to super-resolution tasks. Other applications includes view-synthesis (Mildenhall et al., 2022; Martin-Brualla et al., 2021) and single-image 3D reconstruction (Mescheder et al., 2019; Xu et al., 2019; 2020), by conditioning the inputs on views and images.

Some recent advances have led to structured or hierarchical designs that can further close the gap between generated results and the target function. They divide the complex functions of 3D object and scenes (Chabra et al., 2020; Genova et al., 2020; Jiang et al., 2020) or image (Mehta et al., 2021) into regular subregions, and fitting each subregions individually while considering globally consistency.

As evidenced by Fourier Feature Networks (FFN) (Tancik et al., 2020), ReLU-based MLP networks are struggling to capture highly detailed signals, due to the spectral bias trait. FFN uses positional encoding to map input coordinates of signals to a high dimensional space using sinusoidal functions. Meanwhile, SIREN (Sitzmann et al., 2020) replaces the ReLU activations in the MLP network with sinusoidal functions. They share a similar spirit that manipulates the input or intermediate results in the frequency domain to capture wider frequency bandwidths in the output. In contrast to sinusoidal functions, Spline Positional Encoding (SPE) (Wang et al., 2021) studies learnable spline functions for coordinate embedding. With sufficient local supports of splines, the SPE can also approximate the signal with high frequencies. However, when using a small number of

local supports, the boundaries between patches of the spline become obvious that significantly reduce the visual quality.

Fitting high-frequency details is commonly at the expense of introducing visual artifacts to the results. Structured or hierarchical designs can hardly take effect if a subregion itself is complex. To this end, SAPE (Hertz et al., 2021) adapts a progressive optimization strategy to encode signals with increasing frequencies at individual spatial locations. The method significantly reduces the noisy artifacts, but on the other hand, generates over-smoothed results. The coarse to fine reconstruction strategy has also been used in 3D-aware image synthesis (e.g., BARF (Lin et al., 2021)).

In contrast to the explicit adaptation approaches, i.e., spatial adaptation or structured designs that manipulate the input, we seek an implicit adaptation scheme that is built-in MLPs, directly operates on the continuous functions.

Image Filtering with Deep Neural Networks. Image filters usually compute the weighted average of neighboring pixels of the image as output or sometimes leverage regularization terms for image optimization. Recently, a line of research presents filters with neural networks that can be learned from a large number of datasets for various applications (Jampani et al., 2016; Zhang et al., 2017; Yoon et al., 2015; Gharbi et al., 2017; Chen et al., 2017; Li et al., 2019), such as image super-resolution, denoising, and deblurring, etc. They are mainly based on the convolutional network from which the local neighborhood of the query point is determined. For continuous functions, it is difficult to be aware of the neighbors except to discretize the input domain with limited resolutions.

To enable filtering in the implicit neural representation without sacrificing the global continuousness of the representation is still an open problem. In this work, we seek a building block for filtering in the implicit neural network. The proposed framework has a simple formulation that contains a simple filtering function in the MLPs.

3. Method

In this section, we introduce a new implicit neural network with a filtering function. The framework is illustrated in Figure 1. We first present the implicit network and the positional encoding in Section 3.1. Next, we describe the design of the filtering function in Section 3.2.

3.1. Implicit Networks and Positional Encoding

An implicit neural representation is a function $f_\theta : \mathbb{R}^a \rightarrow \mathbb{R}^n$ which takes as input a coordinate of any query point from the Euclidean space $x \in \mathbb{R}^a$ and predicts a value in the target domain \mathbb{R}^n . An implicit neural network is a parametric model for learning the implicit function f_θ . Examples of the f_θ include mapping pixels to intensity values for image regression or projecting 3D coordinates to scalar values for 3D reconstruction. Training implicit network requires a set of coordinate samplings as input and supervised values as output. Once trained, the network can be sampled to an arbitrarily high spatial resolution, which implies the network has generalizability to unseen space positions beside the training samples.

It is common to implement the implicit network with Multi-Layer Perceptrons (MLPs). However, as evidenced by FFN (Tancik et al., 2020) ReLU-based MLPs tend to generate under-fitting results due to the spectral bias trait. Hence, to avoid the bias of favoring low frequency spectral, FFN projects the input to the frequency domain with controllable frequency bandwidth. Specifically, the position x is represented by a d -dimensional Fourier feature vector, $\gamma(x) \in \mathbb{R}^d$, as follows:

$$\gamma(x) = \frac{s}{\sqrt{d}} [\cos(2\pi x B^T) \| \sin(2\pi x B^T)] \quad (1)$$

where $\|$ is the concatenation of two vectors, and B is an $\frac{a}{2} \times d$ matrix randomly sampled from Gaussian distribution with standard deviation σ . In FFN, s (scale) is set to \sqrt{d} (i.e., the amplification parameter is set to 1 as default), and σ (controls frequency spectrums) is individually searched for different categories. Our network uses a default setting as $s = 5\sqrt{d/2}$, $\sigma = 10$ for all experiments. Note that the scale parameter s has more impact on the convergence speed and would only slightly changes the reconstruction quality.

Fourier feature embedding facilitates learning complex signals of wider frequency bandwidths, however, at the expense of over-fitting issue: introducing unexpected local variations (e.g., noises) to the generated function, especially at unseen positions. As illustrated in Figure 4, a path by linear interpolating two nearby coordinates P_{x_1}, P_{x_2} appears to be overly curvy that consequently yields noisy artifacts to the output. This happens due to the embedding of x_1, x_2 with excessive frequencies from random Fourier features.

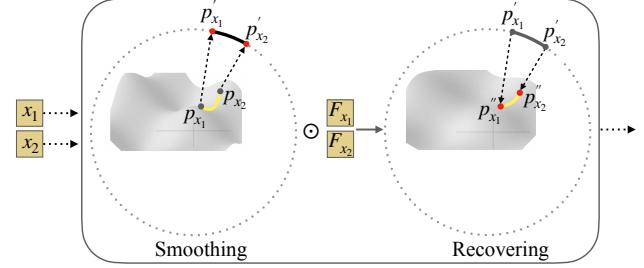


Figure 4. Illustration of smoothing and recovering operations in feature space. The input coordinates x_1, x_2 are projected to features P_{x_1}, P_{x_2} after positional encoding and mapping by a fully connected layer. Note that the Fourier feature space embedded by positional encoding is a hypersphere. After smoothing (i.e. spherical normalization), local variations between them are smoothed out, as shown by the segment between P'_{x_1}, P'_{x_2} within the hyper-sphere. To multiple with initial features F_{x_1}, F_{x_2} , the recovering operation brings P''_{x_1}, P''_{x_2} from the hyper sphere back to a more complex manifold. The interval between the features P''_{x_1}, P''_{x_2} should have less variations than the original features P_{x_1}, P_{x_2} if F_{x_1}, F_{x_2} are chose properly. This process is iterative performed after each fully connected layers.

In general, our work, built upon conventional MLPs (that have under-fitting issues) and the positional encoding (that has over-fitting issues), concerns modulating the generated functions to enable better generalization while achieving high fidelity of neural representations.

3.2. Filtering Function

Let $\theta_i, i = 1, 2, \dots, k$ denote layers of a k -layer MLPs, then the coordinate x passing through the MLPs will produce a sequence of outputs, such as $y_i, i = 1, 2, \dots, k$. To reconstruct the target function, we apply supervision on the final output function y_k . And to avoid generating noisy artifacts, we modulate all the intermediate results with a filtering function. It is also possible to filter the final output directly, but more sophisticated designs are needed to explicitly trade off the data fidelity and the filtering purpose. Practically, we found that filtering on the individual layer of MLPs consequently yields satisfied filtering effects on the final result. The formulation of the new network function can be written in recurrence as follows:

$$\begin{aligned} F_x &= \gamma(x) M^T \\ y_1 &= \theta_1(\gamma(x)) \\ y_i &= o(\theta_i(y_{i-1})) \odot F_x, i = 2, \dots, k-1 \\ f_\theta &= \theta_k(y_{k-1}) \end{aligned} \quad (2)$$

where $o(\cdot)$ is the spherical normalization function, such that $o(v) = \frac{v}{\|v\|}$, and \odot is the Hadamard product. Matrix M maps the $\gamma(x)$ to the same dimensional of the hidden layers.



Figure 5. The influence of the number of hidden layers (or filtering modules) in FINN. In (1), with only one hidden layer, the reconstructed result still contains obvious noisy artifacts. From (2-5), both numerical and visual results are similar. With at least two hidden layers, our method can effectively remove the visual artifacts while keep small patterns, e.g., in the regions of sky and ocean.

When the dimension of $\gamma(x)$ and y_i are the same, M can be ignored.

The filtering function composed of spherical normalization and Hadamard product derives a smoothing and recovering operator respectively that can remove the noisy artifacts effectively while maintaining high fidelity of reconstruction. As illustrated in Figure 4 (left), unexpected local variations occurs between a pair of nearby feature points P_{x_1}, P_{x_2} (in intermediate feature space). After spherical normalization, the linear interpolation between the new points P'_{x_1}, P'_{x_2} becomes flat. This yields an over-smoothed reconstruction, as shown in Figure 3(b), that some small structures are smoothed out. Surprisingly, the image is still better than the one without smoothing operation, e.g., Figure 3(a), even though the filtered function space is a hypersphere and the magnitude of the feature vectors from there have no contri-

bution to the regression. To further restore small structures, the recovering operator is then applied, and the variations between the new features P''_{x_1}, P''_{x_2} are increased, as shown in Figure 4 (right). The generated image by applying both operators is shown in Figure 3(c), where fine details are recovered.

Number of MLP Layers. Our filtering function performs iteratively in the hidden layers of MLPs. In Figure 5, we showcase the performances using a different number of hidden layers, ranging from one layer to five layers. With only one hidden layer (1), visual artifacts are still obvious, and the PSNR and FLIP scores are the worst compared to others. From (2-5), the performance is slightly changed, and the visual quality is quite similar. Even with only one hidden layer, the qualitative and quantitative performance of FINN is still better than state-of-the-art methods.

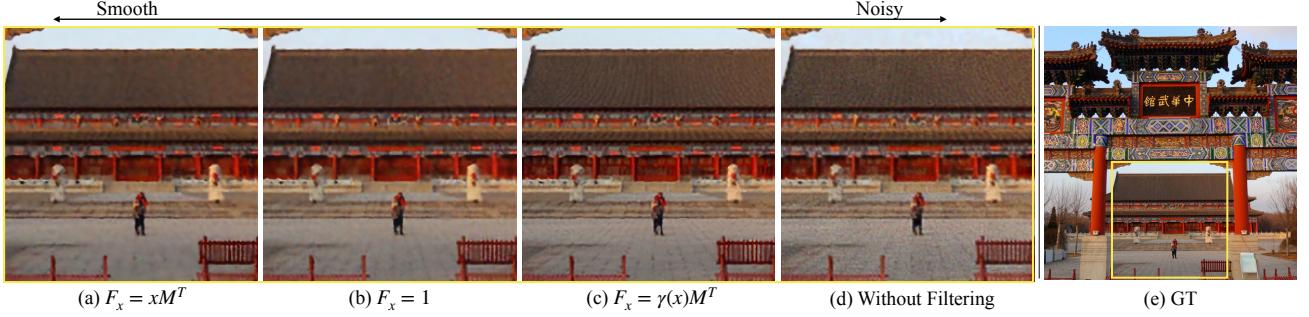


Figure 6. Illustration of filtering effects. F_x is designed in various ways to reach different filtering effects (a-c). Based on the hypersphere, (a) further flattens it by a hyperplane obtained from the linear transformation of the 2D image plane, (b) keeps the space still, and (c) increases local variations brought from $\gamma(x)$.

Filtering Effects. F_x can be properly adapted to reach different recovering effects, as shown in Figure 6. We compare the visual results based on different definitions of F_x , including $F_x = xM^T$, $F_x = 1$, and $F_x = \gamma(x)M^T$ that imply different degree of smoothness. $F_x = \gamma(x)M^T$ is used in FINN, and $F_x = 1$ means the recovering operation is salient. $F_x = xM^T$ directly maps input coordinate to a hyperplane. Thus the recovering information further flattens the local variations from the hypersphere, leading to the smoothest results among all the three variants.

4. Experiments

Our network is applicable for the tasks formulated by the implicit function, focusing on representing complex functions containing a wide range of frequency bandwidths. We validate the benefits of filtering in the implicit neural network on the 2D image regression task.

Our network contains a random Fourier feature mapping and Multi-Layer Perceptrons (MLPs). The MLPs have three hidden layers with ReLU activation and a final layer with Sigmoid Activation. Each hidden layer is associated with a filtering function, as shown in Figure 1. The dimension is 512 for the Fourier feature and 256 for hidden layers. We use MSE loss for training the images. The network is trained for 2000 epoches with a fixed learning rate of 1e-3. For each image, the training pixels are sampled on a regularly-spaced grid containing 25% of the pixels in the image.

We compare our method to state-of-the-art methods, including FFN (Tancik et al., 2020), SIREN (Sitzmann et al., 2020), SPE (Wang et al., 2021) and SAPE (Hertz et al., 2021), on the datasets of natural images and text images from FFN. The error reported are computed on all the pixels in the image, obtained by calculating the similarity between generated RGB values and the ground truth values. We use the standard metrics for similarity measurement, including PSNR and FLIP (Andersson et al., 2020). FLIP gives an

error map for visualization and a global measurement (i.e., weighted median of errors on all pixels).

	PSNR↑		$\text{FLIP} \downarrow$	
	Natural	Text	Natural	Text
FFN	25.57 ± 4.19	30.47 ± 2.11	0.131 ± 0.041	0.096 ± 0.043
SIREN	27.03 ± 4.28	30.81 ± 1.72	0.114 ± 0.040	0.070 ± 0.020
SPE	26.49 ± 3.89	31.12 ± 2.18	0.130 ± 0.038	0.065 ± 0.022
SAPE	28.09 ± 4.04	31.84 ± 2.15	0.118 ± 0.026	0.083 ± 0.041
FINN	28.51 ± 4.35	33.09 ± 1.97	0.100 ± 0.037	0.042 ± 0.016
FINN _{1l}	28.08 ± 4.13	32.18 ± 1.81	0.109 ± 0.038	0.053 ± 0.026

Table 1. Quantitative results on image regression task. FINN_{1l} indicates FINN with one hidden layer.

The numerical results are shown in Table 1. Our method outperforms all state-of-the-art methods with considerable gains. The qualitative results are visualized in Figure 9. In general, all the methods can generate realistic images. In particular, FINN can better capture a wider frequency bandwidth than others, doing better in both backgrounds (low-frequency regions) and small patterns and structures in the images. In contrast, SAPE generates over-smoothed images, while FNN usually includes many noisy artifacts.

Convergence Study. Besides achieving better qualitative and quantitative performance, it is surprising that FINN also has nice properties in both convergence speed and network stability. Figure 8 shows PSNR curves of all the testing images from the natural and text datasets, where FFN (Top) and FINN (Bottom) are both trained for 2000 iterations for comparison. In general, FINN converges after 400 iterations and remains stable after then. In contrast, FNN is sensitive to the SGD-based optimization process, yielding curvy and nonmonotonic PSNR curves. The convergence comparison indicates that the filtering module in FINN serves as similar as a regularization role that significantly alleviates the overfitting issues through the entire optimization procedure.

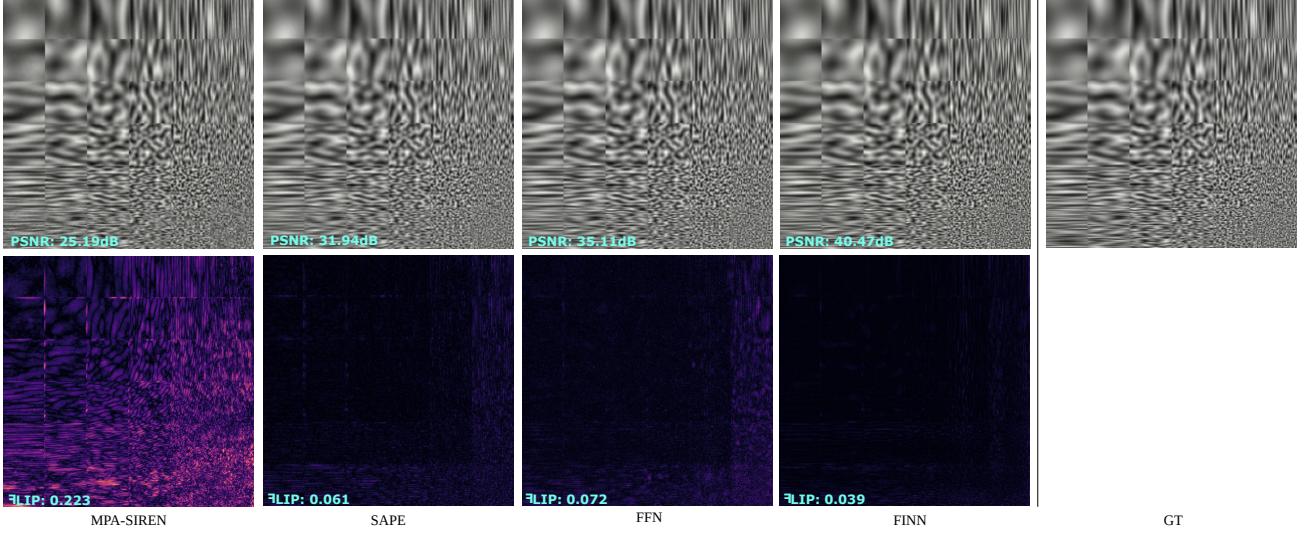


Figure 7. Fitting Perlin noise functions. A Perlin noise image (right) is synthesized with 6×6 small patches with varying frequencies. FINN reconstructs the image with high quality at all frequencies. MPA-SIREN, SAPE, and FFN are struggling to recover high-frequency components. At low-frequency regions, they also generate notable visual artifacts (better visualized by zooming in).

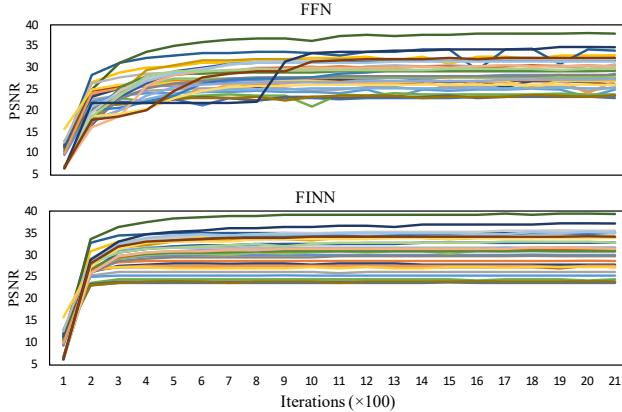


Figure 8. Convergence study. Compared to FFN (Top), FINN (Bottom) converges faster and is much more stable. The PSNR curves painted in different colors indicate different testing images from the natural and text image datasets.

Testing with Wide Frequency Bandwidth Image. We demonstrate the capability of our method in fitting complex signals containing notably wide frequency bandwidth. Similar to MPA-SIREN (Mehta et al., 2021) (not officially named), a testing image is composed of 6×6 grid of Perlin (Perlin, 2002) noise patches with increasing frequencies along with the horizontal and vertical directions. As shown in Figure 7, FINN presents a much better reconstruction compared to MPA-SIREN, SAPE, and FFN. In particular, FINN handles small structures well that other methods usually fail. Even at low-frequency regions, FINN also outperforms compared methods.

5. Conclusion

In this paper, we present a novel implicit neural network with explicit consideration of filtering the generated results. By integrating a filtering module to the implicit neural network, our method can perform data fitting while filtering artifacts. The filtering function is crucial for fitting complex functions that contain a wide range of frequency bandwidths where existing implicit networks can only handle a part of the frequencies, thus leading to either over-smoothing or noisy issues. Our filtering function iteratively performs smoothing and recovering operations to avoid those issues, achieving global smoothness while recovering distinct details.

Experimenting on 2D image regression tasks shows the superiority of our method compared to the state-of-the-art methods. Further, the extended experiment shows that our method has great potential for other images, such as noise images, besides natural and text images. In addition, based on the convergence study, our filtering scheme yields nice properties, such as faster convergence speed and more stable performance under SGD-based optimization.

Acknowledgements

This work was supported in part by National Key R&D Program of China (2018YFB1403901, 2019YFF0302902) and NSF China (61902007).

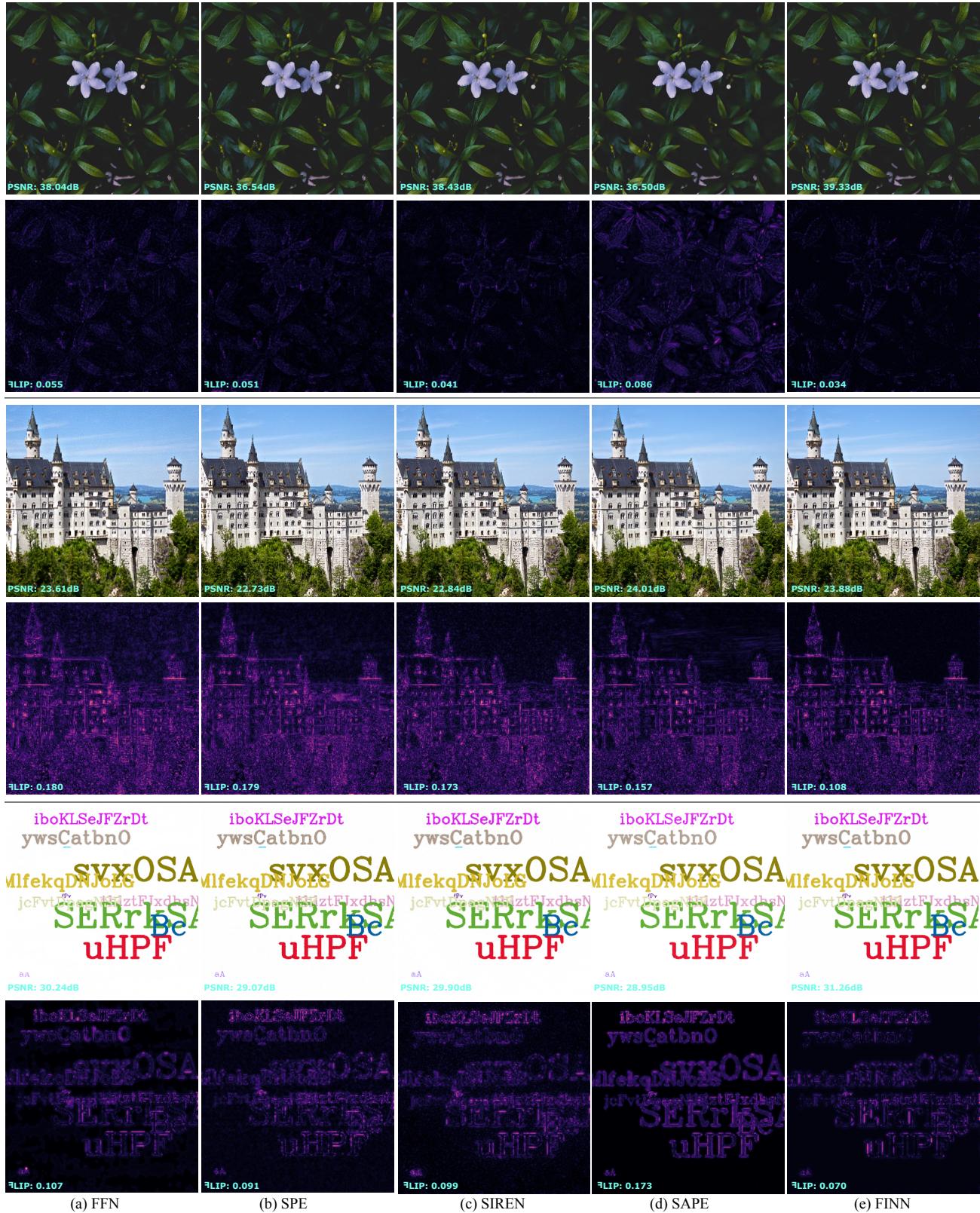


Figure 9. Qualitative results on image regression task. Better visualized by zooming in.

References

- Andersson, P., Nilsson, J., Akenine-Möller, T., Oskarsson, M., Åström, K., and Fairchild, M. D. FLIP: A Difference Evaluator for Alternating Images. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 3(2):15:1–15:23, 2020.
- Atzmon, M. and Lipman, Y. SAL: sign agnostic learning of shapes from raw data. In *CVPR*, pp. 2562–2571. Computer Vision Foundation / IEEE, 2020.
- Chabra, R., Lenssen, J. E., Ilg, E., Schmidt, T., Straub, J., Lovegrove, S., and Newcombe, R. A. Deep local shapes: Learning local SDF priors for detailed 3d reconstruction. In *ECCV (29)*, volume 12374 of *Lecture Notes in Computer Science*, pp. 608–625. Springer, 2020.
- Chen, Q., Xu, J., and Koltun, V. Fast image processing with fully-convolutional networks. In *ICCV*, pp. 2516–2525. IEEE Computer Society, 2017.
- Chen, Z. and Zhang, H. Learning implicit fields for generative shape modeling. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- Genova, K., Cole, F., Sud, A., Sarna, A., and Funkhouser, T. A. Local deep implicit functions for 3d shape. In *CVPR*, pp. 4856–4865. Computer Vision Foundation / IEEE, 2020.
- Gharbi, M., Chen, J., Barron, J. T., Hasinoff, S. W., and Durand, F. Deep bilateral learning for real-time image enhancement. *ACM Trans. Graph.*, 36(4):118:1–118:12, 2017.
- Hertz, A., Perel, O., Giryes, R., Sorkine-Hornung, O., and Cohen-Or, D. Sape: Spatially-adaptive progressive encoding for neural optimization. *arXiv preprint arXiv:2104.09125*, 2021.
- Jampani, V., Kiefel, M., and Gehler, P. V. Learning sparse high dimensional filters: Image filtering, dense crfs and bilateral neural networks. In *CVPR*, pp. 4452–4461. IEEE Computer Society, 2016.
- Jiang, C. M., Sud, A., Makadia, A., Huang, J., Nießner, M., and Funkhouser, T. A. Local implicit grid representations for 3d scenes. In *CVPR*, pp. 6000–6009. Computer Vision Foundation / IEEE, 2020.
- Li, Y., Huang, J., Ahuja, N., and Yang, M. Joint image filtering with deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(8):1909–1923, 2019.
- Lin, C., Ma, W., Torralba, A., and Lucey, S. BARF: bundle-adjusting neural radiance fields. *CoRR*, abs/2104.06405, 2021.
- Martin-Brualla, R., Radwan, N., Sajjadi, M. S. M., Barron, J. T., Dosovitskiy, A., and Duckworth, D. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, pp. 7210–7219. Computer Vision Foundation / IEEE, 2021.
- Mehta, I., Gharbi, M., Barnes, C., Shechtman, E., Ramamoorthi, R., and Chandraker, M. Modulated periodic activations for generalizable local functional representations. *CoRR*, abs/2104.03960, 2021.
- Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. Occupancy networks: Learning 3D reconstruction in function space. In *Proc. CVPR*, 2019.
- Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. Nerf: representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1):99–106, 2022.
- Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S. DeepSDF: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019.
- Perlin, K. Improving noise. *ACM Trans. Graph.*, 21(3): 681–682, 2002.
- Sitzmann, V., Zollhöfer, M., and Wetzstein, G. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *NeurIPS*, pp. 1119–1130, 2019.
- Sitzmann, V., Martel, J. N., Bergman, A. W., Lindell, D. B., and Wetzstein, G. Implicit neural representations with periodic activation functions. In *Proc. NeurIPS*, 2020.
- Tancik, M., Srinivasan, P. P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J. T., and Ng, R. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020.
- Wang, P., Liu, Y., Yang, Y., and Tong, X. Spline positional encoding for learning 3d implicit signed distance fields. In *IJCAI*, pp. 1091–1097. ijcai.org, 2021.
- Xu, Q., Wang, W., Ceylan, D., Mech, R., and Neumann, U. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *arXiv preprint arXiv:1905.10711*, 2019.
- Xu, Y., Fan, T., Yuan, Y., and Singh, G. Ladybird: Quasi-monte carlo sampling for deep implicit field based 3d reconstruction with symmetry. In *ECCV (1)*, volume 12346 of *Lecture Notes in Computer Science*, pp. 248–263. Springer, 2020.

Yoon, Y., Jeon, H., Yoo, D., Lee, J., and Kweon, I. S. Learning a deep convolutional network for light-field image super-resolution. In *ICCV Workshops*, pp. 57–65. IEEE Computer Society, 2015.

Zhang, J., Pan, J., Lai, W., Lau, R. W. H., and Yang, M. Learning fully convolutional networks for iterative non-blind deconvolution. In *CVPR*, pp. 6969–6977. IEEE Computer Society, 2017.