Neural Fourier Filter Bank

(Supplementary Material)

A. Experiment details

A.1. More details about Fourier grid features

In Sec. 3.1 of the main text, the multi-level Fourier grid features are defined to compute the continuous mapping between the input coordinate $\mathbf{x} \in \mathbb{R}^n$ and the m dimension feature space. Following instantNGP, we set the base resolution N_{min} and a scaling coefficient c_g between adjacent levels to define the resolution for a certain level l as:

$$N_l = N_{min} \cdot c_g^l, \tag{11}$$

where the level index l starts from 0. We adjust the total number of levels to balance the ability to model fine details and the complexity of the model itself. To compute the variance values that we use to initialize the Fourier features, we apply a similar scaling strategy:

$$\sigma_l = \sigma_{min} \cdot c_f^l, \tag{12}$$

where σ_{min} and c_f represent the base variance value and its corresponding scaling coefficient. Roughly, we set $\sigma_{min} = \sqrt[2]{N_{min}}$ and $c_g \approx c_f$. However, the optimal choice of these values is circumstantial and we modify N_{min}, c_g, σ_{min} and c_f for each task.

A.2. 2D image fitting

To roughly match the model capacity used by other methods, for the 'Tokyo' image we use fully-connected layers with 96 neurons, and for the 'Einstein' image 256. All fully connected layers are using sine activations as previously described in the main text. Additionally, for the 'Tokyo' image we use $N_{min} = 64$, $c_g = 1.5$, $\sigma_{min} = 5.0$ and $c_f = 2.0$, and for the 'Einstein' image we use $N_{min} = 64$, $c_g = 2.0$, $\sigma_{min} = 10.0$ and $c_f = 2.0$.

For both images, we train our network with 50,000 iterations to ensure full convergence—our method already converges after 20,000 iterations. To well-reconstruct complex high-frequency signals, we set α_i in Eq. (5) to 100.

A.3. 3D SDF regression

For this task, we train our network for 50,000 iterations on 26 million sampled points with a batch size of 49,152, to maximize GPU memory utilization. As the SDF has varying level-of-detail—*e.g.*, smooth regions can be very smooth, while detailed regions can have high-frequency detail—we set the number of levels to five for the Fourier grid feature. For each level, we use fully-connected layers with 256 neurons. We further set $N_{min} = 8$, $c_g = 1.3$, $\sigma_{min} = 5$ and $c_f = 1.2$. For both shapes in Tab. 2, we choose $\alpha_i = 45$, which we empirically found to provide the best balance between high- and low-frequency details for this task.

	$T = 2^{17}$			$T = 2^{19}$			$T = 2^{21}$		
	$L = 8$	$L = 10$	$L = 12$	$L = 8$	$L = 10$	$L = 12$	$L = 8$	$L = 10$	$L = 12$
InstantNGP	28.18	30.89	31.93	29.38	33.37	36.41	30.41	36.37	41.28
Ours	30.31	32.86	33.82	31.28	34.53	37.56	31.43	36.86	41.36

Table A. **Performance under varying T and L** – Our method shows higher PSNR values for 'Tokyo' image with various T and L settings.

A.4. Neural radiance field

For this task, we closely follow the experimental setup of InstantNGP, including the four levels for the grid. For our method, we use 128 neurons to match a similar model capacity as the baseline. We further set $\alpha_i = 20.0$, $N_{min} = 64$, $c_g = 2.0$, $\sigma_{min} = 8.0, c_f = 1.4$ to balance model complexity and synthesis quality.

A.5. Preparing SDF data for SDF regression

To obtain the ground-truth SDF values, we use pysdf[1]. We use the original mesh files and normalize them into a unit sphere to standardize shapes. When training each model, for each batch, we sample 49152 points for training where 20% of the points are sampled uniformly within the volume, 30% of the points are sampled near the shape surface, and the rest are sampled directly on the surface.

A.6. Experimental setting for InstantNGP

Generally, our choices are based [37, Sec. 3.]. As shown in [37, Fig. 5.], F=2 and L=16 are good choices for the feature dimension F and feature level L. For the hash table size T, we choose 2^{19} as it is when the performance starts being throttled as shown in [37, Fig. 4.]. For the 'Einstein' image in Tab. 1 of the main text, we reduced the model's capacity as the image is simpler.

In addition, we use various settings for T and L for both InstantNGP and our method and report the results for 'Tokyo' image in Tab. A. Regardless of the hyperparameter settings, our method outperforms InstantNGP consistently.

B. More ablation studies

As discussed in Sec. 3 of the main text, our key idea is the Fourier grid features, and the wavelet-inspired composition. Here, we further justify our design choices based on experiments.

[1]Github link: https://github.com/sxyu/sdf

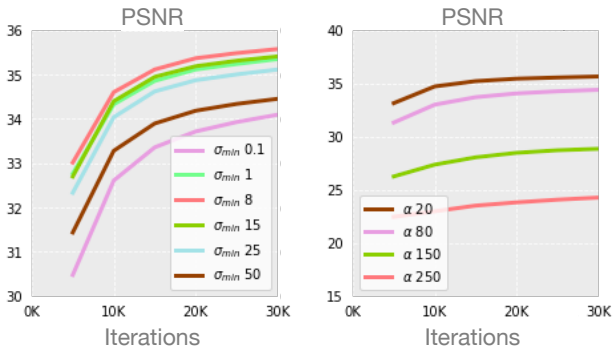

Figure A. Ablation studies for the Fourier feature variance (left) and the scaling factor α in fully-connected layers (right). Both parameters highly affect how frequency is dealt with within our framework, and thus require optimal values to be set. These parameters are mostly task dependant.

The effect of grid resolutions. In Sec. 1., we discuss how grid resolution relates to what frequency range a model can reconstruct. In Fig. B, we illustrate that this is indeed the case by varying N_{min}. Also in Fig. C, we show how the scaling factor c_g affects final results. As expected, whether fine details are preserved or not depends highly on the two parameters.

The effect of the Fourier feature variance. In Sec. 3.1, we discuss how our initialization strategy leads to the natural biasing of frequency components. Thus, this variance has a strong impact on the performance of the method—too high variance would lead to the method focusing only on high-frequencies, while too low variance would cause the opposite. Thus, this variance should be selected with care. In Fig. A, we show how the variance σ_{min} affects the final reconstruction performance—σ_{min} should roughly be in a proper range, as demonstrated by the $\sigma_{min} = 1$ and $\sigma_{min} = 8$ results.

The effect of the scaling factor α_i. Similarly, α_i is another parameter that highly impacts how each layer combines grid features and the features from the previous layer. We set a single global value for all layers for simplicity, and experiment with multiple values to demonstrate its effect in Fig. A. As expected, a properly tuned value is necessary for optimal performance. We found this parameter to be highly task dependant.

The effect of Fourier encodings We also demonstrate the influences of applying Fourier encodings to the low dimensional grid features by only preserving the Grid+MLP components. We train this ablated model for the 'Tokyo' image which gives the PSNR of 30.48, whereas the full model yields 31.57. To further evaluate the effects of activations for grid features, we implement by replacing the sine activation functions with Relu and produce 30.39, highlighting the necessity of current design choices.

	2D Fitting				3D Fitting		
	Size (MB)↓	PSNR↑	SSIM↑	LPIPS↓	Size (MB)↓	F-score↑	CD↓
InstantNGP [33]	36.0	37.93	0.9578	0.092	46.5	0.845	0.00295
SIREN [41]	5.2	33.35	0.9227	0.253	2.0	0.806	0.00370
ModSine [30]	3.5	28.63	0.8316	0.409	12.0	0.604	0.00386
Ours*	4.1	34.64	0.9326	0.136	-	-	-
Ours	10.0	38.64	0.9672	0.064	1.4	0.833	0.00297

Table B. **More comparisons** – With more compact networks, our method can produce competitive or better results compared to baselines. Our smaller model (Ours*) is achieved by using smaller grid sizes.

The effect of fully-connected layer size. The size of the MLP also plays an important role, as it allows for more complex composition of signals coming from different frequencies. In Fig. D, we illustrate the importance of the MLP size— the larger the better, but with an increase in computation and model complexity.

The effect of the Fourier grid level. Finally, we demonstrate how the number of Fourier grid levels affects our results. As expected, we observe in Fig. E that the models with higher grid levels consistently provide better results.

C. More visualization results

In Fig. G and Fig. H, we provide more detailed look into the 2D reconstruction results. Both results provide highly impressive reconstructions, without any discernable differences to the ground truth.

In Fig. F, we further provide the qualitative results for regressing the 'Asian Dragon' shape SDF. Our method and InstantNGP both provide results with very fine details, but ours is more compact.

We provide more qualitative results for novel view synthesis in Fig. I. As shown, our method is able to provide synthesis results with both low-frequency details as shown by the 'Lego' scene, and high-frequency details as shown by the 'Ficus' scene with thin structures.

D. More comparison results

While hyperparameters differ for each task, we found them to be generally applicable to other scenes for the same task. In Tab. B with the hyperparameters used in the main text, we compare our method on six high-resolution images with each image having more than 10 million pixels, and ten 3D scenes with complicated geometric details. It is clear that the proposed method can consistently achieve better or comparable results with much smaller model size.

E. Note on comparison with ModSine

We use the local representation with a tile size of 64 for 2D && 3D signal fitting, under the auto-decoding setup. For ModSine [34], we have taken the network from the of-

ficial implementation[2] and included it in our training and evaluation code, to keep all training aspects identical to ours. We note, however, that our results might not have optimal hyperparameter settings, as some of the experimental setups (layer number, layer size, batch size, and learning rate) were chosen by us as they were unavailable in [34].

F. Discussions about runtime

As shown in Tab. 3 of the main text, our current implementation is not utilizing CUDA libraries (e.g. tiny-cuda-nn[3]) in places other than the hash grid, thus slower than InstantNGP as of now. Our current implementation requires around 13 minutes to train a Blender scene for the NeRF task, whereas InstantNGP takes around 3–4 minutes. However, we suspect that with a more efficient implementation, for example with a full CUDA-integrated implementation such as InstantNGP, would greatly accelerate our method, as our method only introduces a few small linear layers and Fourier Feature embedding layers, which should not increase the computation load significantly. Finally, recall that as shown in Fig. 1 of the main paper, our method converges faster in terms of number of optimization steps than other methods, including InstantNGP.

G. Limitations and future work

One limitation of our work is that we assume a stationary neural field, which is not conditioned, similar to how InstantNGP is limited. Thus, a potentially fruitful research direction would be to incorporate recent conditional neural field methods into our framework. We also notice that all grid-based methods do have issues when modeling very large scenes. This is also another potentially interesting research direction.

H. Broader impact

Our work is of fundamental nature and is not immediately linked to any particular application. However, our method would facilitate efficient neural field representations, which can widen the potential application area of neural fields. In addition, our method, being more efficient, would reduce the amount of computing and power consumption required for the application of these methods.

[2]Code from https://ishit.github.io/modsine/
[3]Code from: https://github.com/NVlabs/tiny-cuda-nn

Figure B. The ablation study for the base resolution N_{min}. With larger N_{min}, fine details are better preserved.

Figure C. The ablation study for the scaling coefficient c_g. With larger c_g, reconstruction quality improves, with more fine details being preserved and with higher spatial resolution.

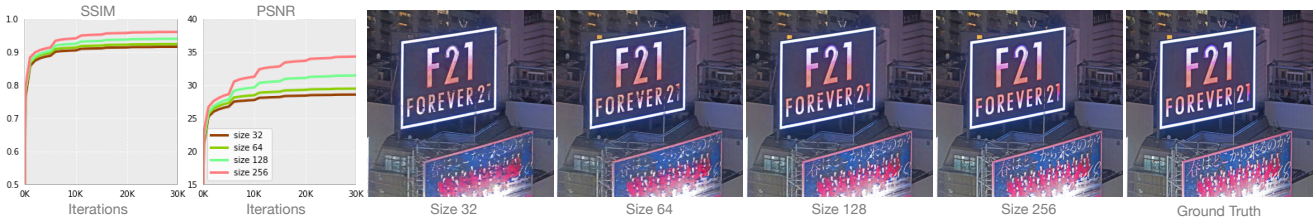

Figure D. The ablation study results for the MLP size. As the MLP size increases, the network becomes better at composing signals from various levels, thus various frequencies, leading to a better final outcome.

Figure E. The ablation study for the number of levels for the Fourier grid feature. More levels lead to a drastic increase in the quality of fine details.

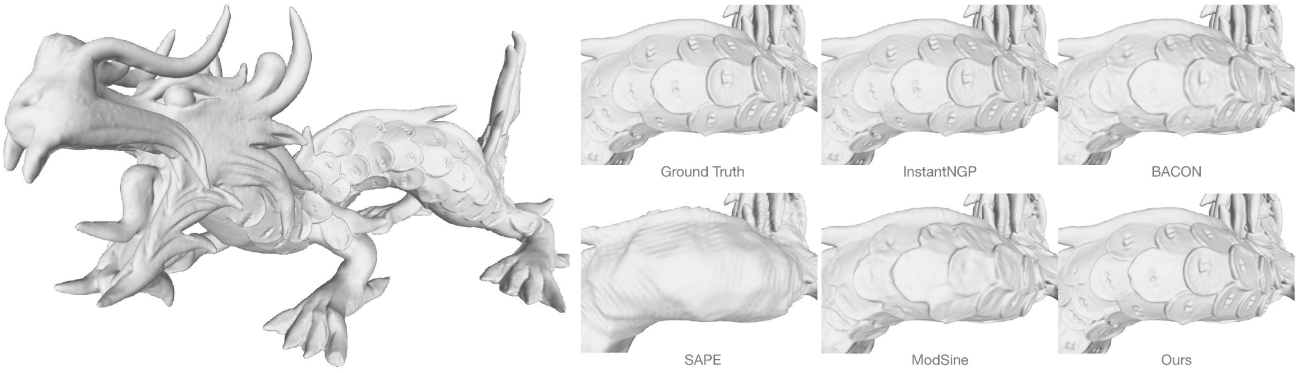

Figure F. 3D fitting result for the 'Asian Dragon'. The left sub-image is the ground truth shape while six zoomed insets are shown on the right for better detail visualizations.

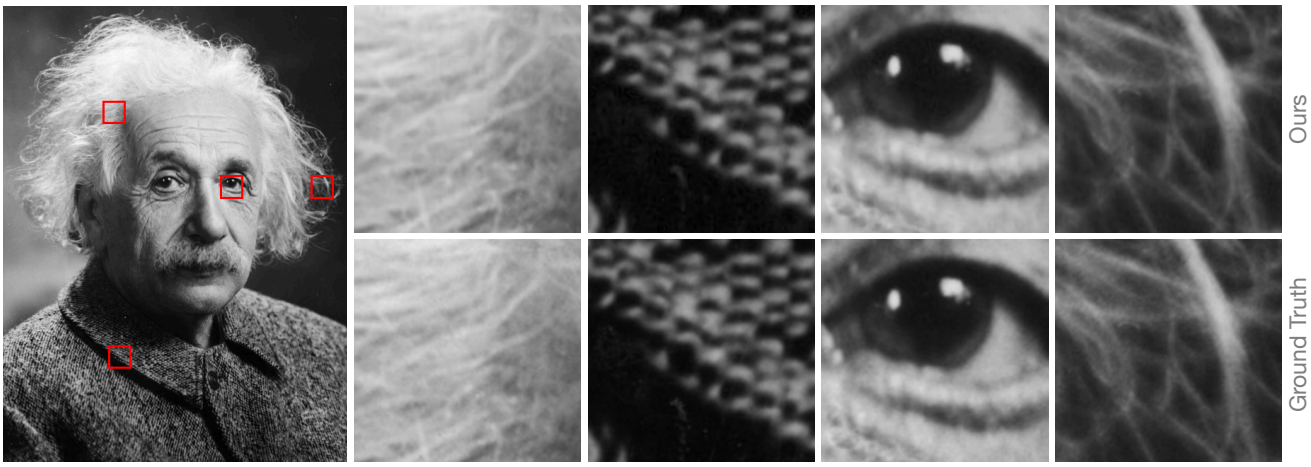

Figure G. 2D fitting result for 'Einstein' image. Our entire reconstructed image is presented on the left while four close-up views are presented on the right. Note how our reconstructions are near-perfect for both coarse and fine details.

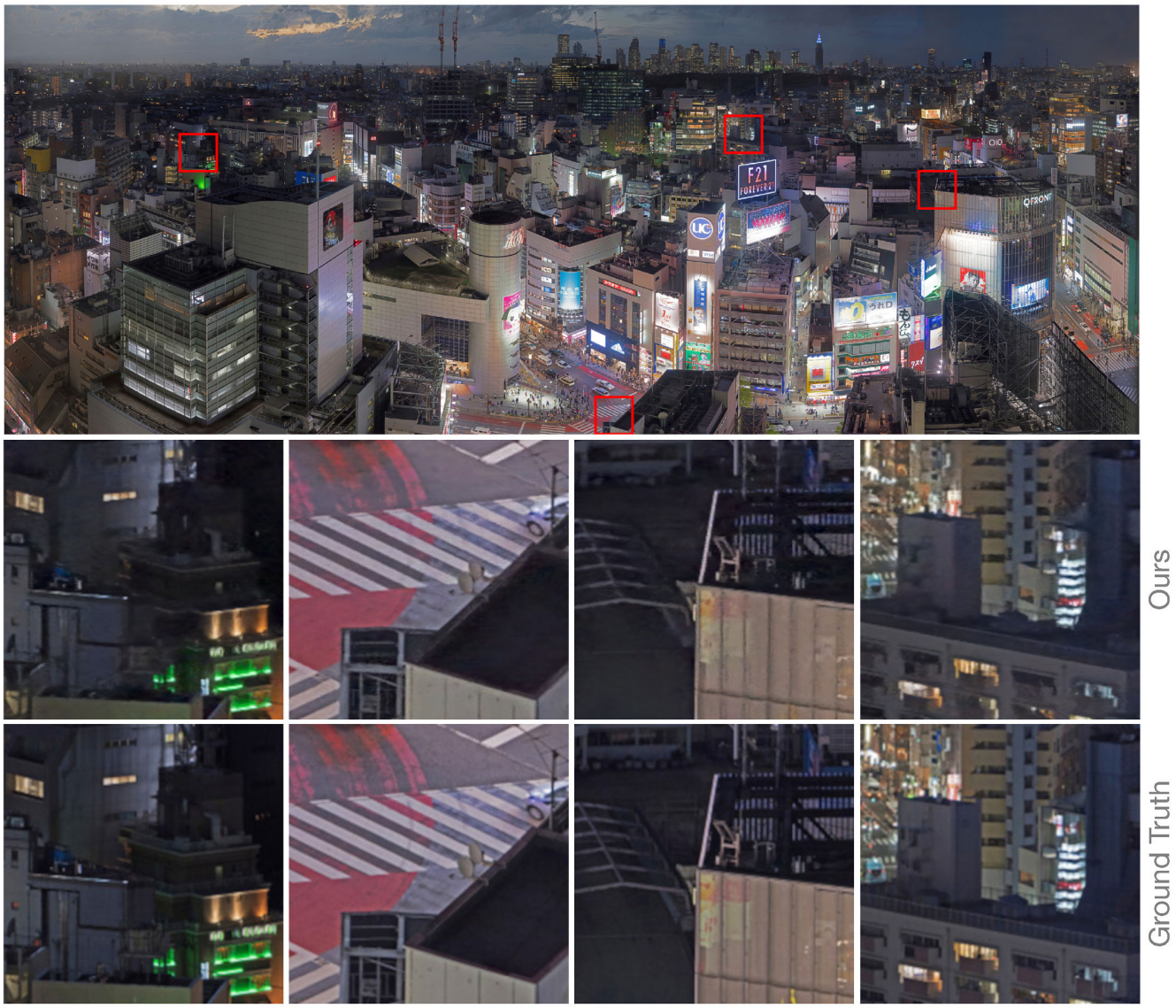

Figure H. 2D fitting result for 'Tokyo' image. Our entire reconstructed image is presented on the top while four close-up views are presented on the bottom. Our method provides near-perfect reconstruction.

Figure I. Qualitative results for novel view synthesis with neural radiance fields. Our method is able to clearly reconstruct the textures (*e.g.*, the chair on 2^{nd} row) and the geometric details (*e.g.* the lego on the last row).