Positional-Encoding Image Prior

Nimrod Shabtay*, Eli Schwartz*, Raja Giryes
Tel-Aviv University

Abstract

In Deep Image Prior (DIP), a Convolutional Neural Network (CNN) is fitted to map a latent space to a degraded (e.g. noisy) image but in the process learns to reconstruct the clean image. This phenomenon is attributed to CNN’s internal image prior. We revisit the DIP framework, examining it from the perspective of a neural implicit representation. Motivated by this perspective, we replace the random latent with Fourier-Features (Positional Encoding). We empirically demonstrate that the convolution layers in DIP can be replaced with simple pixel-level MLPs thanks to the Fourier features properties. We also prove that they are equivalent in the case of linear networks. We name our scheme “Positional Encoding Image Prior” (PIP) and exhibit that it performs very similar to DIP on various image-reconstruction tasks with much fewer parameters. Moreover, we show that PIP can be easily extended to videos, where image-prior and INR methods struggle and suffer from instability. Code and additional examples for all tasks, including videos, are available on the project page nimrodshabtay.github.io/PIP.

1 Introduction

Deep Image Prior (DIP) [36] has shown that if a CNN is trained to map random noise or a learned latent code to a degraded image, it will either converge to the restored image (e.g. for super-resolution) or produce the restored image in the middle of the optimization process (e.g. for denoising). This has been attributed to an internal image prior that CNNs have. Even though DIP is used in a zero-shot setup with no supervision and with only a degraded image available, it achieves impressive results for many image restoration tasks, including denoising, super-resolution, inpainting, and more. DIP was also extended to other tasks such as segmentation and dehazing using DoubleDIP [9], video processing [44] and 3D mesh reconstruction [12].

Despite the remarkable success of DIP, it is still unclear why fitting random noise to a deteriorated image can restore the image. One explanation of how DIP works, but not why it works, is that the CNN learns to fit first the low frequencies and only later the higher frequencies, hence early stopping has a low-pass filter effect [33].

In this work, we propose that DIP should be considered as a neural implicit model that is trained to represent the target image. These models gained high popularity in the task of 3D scene representation, due to their use in Neural Radiance Fields (NeRF) [26].
and its many follow-up works. They were also used for 2D image super-resolution [4], but not in the zero-shot form, where they learned to map representations from a pre-trained CNN to RGB values.

Neural implicit functions are neural models trained to represent a signal, by training a multilayer perceptron (MLP) to map the coordinates of the target signal (an image in our case) to the value in these coordinates (RGB color in the case of images). This results in a neural network that implicitly represents the image, i.e., it can generate it by simply calculating the output of all its coordinates and even interpolating the image by using a finer scale of the coordinates.

Training an implicit model with the raw coordinates as input results in an oversmoothed output due to the spectral bias of neural networks towards low frequencies [30]. To overcome that, it has been shown that by mapping the input coordinates to Fourier-Features (FF), the implicit model can learn higher frequencies. This leads to a remarkable improvement in the ability of neural implicit representations.

Taking the neural implicit representation perspective, we may think of the inputs of DIP as the equivalent of the coordinate representation in implicit models. With this view in mind, DIP maps random codes to RGB values. However, despite the initial impression, this is not totally a random code. Providing independent random codes for each coordinate and then training an implicit neural model using MLP for a given image simply does not work. Yet, due to the convolutional structure of DIP, there is a “structure” in its randomness. The encoding for a specific location in the image is a shifted version of the random codes of its neighboring coordinates. Moreover, for a linear network, we prove that MLP with FF is equivalent to CNN with random inputs. Thus, DIP can be considered as an implicit function that maps shifted random codes of the coordinates to the RGB values of the target image. With this view in mind, a natural question is why not simply use other types of positional encodings for the input coordinates? Or more specifically, Fourier features, which have been proven to be successful for neural implicit representation.

We refer to this setup of DIP with positional encoding inputs as “Positional Encoding Image Prior” (PIP). In most cases, PIP-CNN outperforms DIP-CNN in image-restoration tasks such as denoising and super-resolution. Moreover, we show that using Fourier features the convolutional layers in the DIP architecture can be replaced by MLP operating independently on each coordinate (implemented by $1 \times 1$ conv.). The MLP-based architecture is inspired by the MLP used for implicit models, but it is not identical as we still have skip-connection and down/up-sampling like in the original DIP. We denote this network structure as PIP-MLP, and show that it achieves similar performance compared to conventional DIP-CNN but with a lower parameter count and FLOPs. Note that using DIP-MLP fatally fails.

A remarkable advantage of PIP is that it can easily be adapted to other modalities, such as video (3D), by modifying its encoded dimensions. In contrast, extending DIP to video by employing random input encoding with 3D convolutions (3D-DIP) struggles to give reliable results. PIP, on the other hand, achieves high-quality reconstruction and significantly outperforms image-prior and INR methods for video tasks using a simple extension of 2D PIP to 3D and using Fourier features that encode both spatial and temporal dimensions.

To summarize, the contributions of our paper are threefold: (i) a novel interpretation of DIP; (ii) an efficient network for self-training; and (iii) a significant improvement in video denoising and super-resolution from a single video.

2 Related Work

Positional encoding was first presented in the transformer architecture [37], where the importance of positional encoding in learning without an explicit structure was demonstrated. The visual transformer (ViT) [6] extended the transformer architecture along with the concept of positional encoding to the 2D visual domain, where an image is broken into patches and each patch contains the pixel values along with their corresponding spatial location in the image. Fourier-features were used as the positional encoding of transformers for achieving accurate segmentation
masks in medical imaging applications [28].

Another area of research utilizing positional encoding is implicit neural functions, which are neural networks optimized to map input coordinates to target values. It was demonstrated in [35] that representing the input coordinates as Fourier-features with a tuneable bandwidth enables a simple MLP to generate complex target domains such as images and 3D shapes while preserving their high-level details. In Neural Radiance Fields (NeRF) [27], implicit functions with Fourier features are used for synthesizing novel views of a 3D scene from sparse 2D images. SIREN [34] showed that by changing the activation function of a simple MLP to a periodic one, the network can represent the spatial and temporal derivatives so one can use a regular grid of input coordinates to successfully recover a wide range of target domains (images, videos, 3D surfaces, etc.). Local implicit image functions (LIIF) [4] represent an image as a continuous function using an implicit model. This allows it to perform super-resolution at an arbitrary scale. Xu et al. [39] show how positional encoding can be interpreted as a spatial bias in GANs. SAPE [14] and BACON [22] demonstrate how a combination of a coordinate network along with a limitation of its frequency spectrum can achieve a multi-scale representation of the target domain. Due to the success of implicit models, coded by Fourier-features, other areas of research incorporate Fourier-features. For example, Li et al. [20] apply learned Fourier-features to control the inputs to gain better sample efficiency in many reinforcement learning problems. [16] proposed to use implicit neural representation with a weight regularization for each layer for blind image restoration. Bustos-Brinez et al. [2] use random Fourier-features to perform accurate data-set density estimation for anomaly detection. This work treats DIP as an implicit model showing that as such it can be used with FF and an MLP.

Deep Image Prior (DIP) [36] showed that a CNN network (typically a U-net shaped architecture) can recover a clean image from a degraded image in the optimization process of mapping random noise to a degraded image. The power of DIP is shown for several key image restoration tasks such as denoising, super-resolution, and inpainting. DIP was extended to various applications. Double-DIP [9] introduced a system composed of several DIP networks, where each learns one component of the image such that their sum is the original image. It has been used for image dehazing and foreground/background segmentation. SelfDeblur [32] performed blind image deblurring by simultaneously optimizing a DIP model and the corresponding blur kernel. In the medical imaging domain, several works tried to utilize the power of image priors to reconstruct PET images [40, 13, 10, 11]. Other works tried to improve the performance of image priors using modifications and additions to the original setup. Mataev et al. [24] and Fermanian et al. [7] combined DIP with the plug-and-play framework to improve DIP’s performance in several inverse problems. Jin et al. [15] extended DIP to blended distortion priors. Zukerman et al. [45] showed how the back-projection fidelity term can improve DIP performance in restoration tasks like deblurring. In other areas, Kurniawan et al. [17] proposed a method for demosaicing based on DIP. Chen et al. [5] suggested DIP-based neural architecture search.

| Method       | σ = 25 | σ = 10 | σ = 5 | σ = 10 Poisson | SR x4 | SR x8 |
|--------------|--------|--------|-------|----------------|-------|-------|
| SIREN [44]   | 27.94  | 29.78  | 29.6  | 29.6           | 27.66 | 24.18 |
| Titan [19]   | 25.11  | 25.20  | 25.21 | 21.6           | 17.43 |
| PIP          | 25.35  | 25.83  | 25.54 | 24.8           | 21.33 |

Table 1: Denoising and Super-Resolution image-prior evaluation (PSNR ↑). Compared to DIP-CNN, DIP-MLP fails. On the other hand, PIP is robust to the model used and does not require a CNN. Using fixed frequencies is slightly better for denoising, while learned frequencies (optimizing input frequencies together with model params) are slightly better for SR.

| Method     | σ = 25 | σ = 10 | σ = 5 | σ = 10 Poisson | SR x4 | SR x8 |
|------------|--------|--------|-------|----------------|-------|-------|
| SIREN [44] | 27.94  | 29.78  | 29.6  | 29.6           | 27.66 | 24.18 |
| Titan [19] | 25.11  | 25.20  | 25.21 | 21.6           | 17.43 |
| PIP        | 25.35  | 25.83  | 25.54 | 24.8           | 21.33 |

Table 2: Denoising and Super-Resolution implicit models evaluation (PSNR ↑). Compared to other implicit models, PIP performs better or similarly compared to all new proposed methods.
1.3dB), but not as significant as for DIP. In performance for PIP on SR without the U-Net (- its performance for denoising. We observe a drop convolutions and the U-Net structure, PIP retains its performance for denoising. We observe a drop in performance for PIP on SR without the U-Net (-1.3dB), but not as significant as for DIP.

Table 3: The effect of the U-net structure is evaluated on both the denoising and super-resolution tasks (PNSR ↑). While DIP is ineffective without convolutions and the U-Net structure, PIP retains its performance for denoising. We observe a drop in performance for PIP on SR without the U-Net (-1.3dB), but not as significant as for DIP.

DIP [36] demonstrates that if an untrained CNN model is fitted, in a zero-shot manner, to a corrupted image, it will eventually recover the clean image without adding any explicit regularization. Formally, we have $x = f_0(z)$, where $f_0$ is a neural network that maps random input codes $z \in \mathbb{R}^{C \times W \times H} \sim U(0, \frac{1}{10})$ to image space $x \in \mathbb{R}^{3 \times W \times H}$. Given a corrupted image $x_0$, we want to recover its clean version. Such an image reconstruction task can be formulated as an energy minimization problem

$$x^* = \arg\min_x E(x; x_0) + R(x),$$

where $E(x; x_0)$ is the data term and $R(x)$ is a regularization term. The main observation in DIP is that, due to the CNN’s image prior, the regularization term can be dropped and the data term can be optimized by adapting the model parameters with gradient descent

$$\theta^* = \arg\min_{\theta} E(f_0(z); x_0), \quad x^* = f_{\theta^*}(z).$$

As illustrated in Fig. 1, one may consider DIP as an implicit model and its random input as positional encoding of shifted random patches. Motivated by the above relations, in PIP, we suggest replacing the random input codes $z$ with Fourier-features as the positional encoding of the image coordinates. These features are advantageous of being smooth, continuous, and with controllable bandwidth. For a coordinate vector $v$ and a frequency set $\{f_0, \ldots, f_{m-1}\}$, the Fourier-features are

$$\gamma(v) = [\ldots, \cos(f_i v), \sin(f_i v), \ldots].$$

We use log-linear spaced frequencies $f_i = \sigma^{i/m}$, with the base $\sigma$ set by a predefined maximum frequency, such that $f_{m-1} = f_{\text{max}}$. (see 15 for a visual diagram.

3 Positional-Encoding Image Prior (PIP)

Table 4: Activation effect on reconstruction results. We evaluate (PSNR ↑) recent proposed activations for implicit models such as Sine and Gaussian and compare them to the originally proposed Leaky-ReLU over several image reconstruction tasks. Leaky-ReLU outperformed all activations.

| Denoising       | SR ×4 | SR ×8 |
|-----------------|-------|-------|
| Gaussian        | 28.26 | 29.84 | 27.87 | 22.35 |
| Sine            | 27.21 | 28.12 | 25.85 | 20.80 |
| Leaky-ReLU      | 28.26 | 29.78 | 25.95 | 19.76 |

Figure 3: SR examples - First example is for ×4 SR; Second is for ×8 SR. The results for DIP (CNN) and PIP (MLP) are very similar, suggesting they have a similar image prior.

A recent work [19] tried to draw a connection between image priors and implicit neural representations. While they mentioned the connection, it is not complete with a partial comparison. PIP comprehensively draws a connection to implicit models both analytically and through a comprehensive study.
Figure 4: SIREN vs. PIP frequency hyper-parameter effect. We apply different Band-Width limits to SIREN (different \(\omega_0\) values) and evaluate the reconstruction results over the "standard dataset". PIP produce over-smooth images when the max frequency is low but converge to a steady plateau as the frequency range gets higher as oppose to SIREN where we can see an optimal point around \(\omega_0 = 2^5 = 32\) but the performance drops as the frequency range move away from the optimal point.

of the input encoding. Inspired by [21], we also tested a learnable-frequencies variant, where the frequencies are initialized as described above and then optimized together with the model parameters using gradient descent. This variant allows learning the most appropriate encoding to be used for a given image and is beneficial for some tasks.

We further suggest replacing the convolutional layers with an element-wise optimization with a coordinate-MLP, which is equivalent to performing a 1 \(\times\) 1 convolution on the whole input. The intuition is that the PE image prior is sufficient to make the convolution redundant. Moreover, in the following proposition, we show that for a linear model, DIP can be formulated as learning of an implicit function with Fourier features as its input. Full details are in Appendix A.

**Proposition 1.** For \( E \) being the \( \ell_2 \) loss and \( f(z) = h * z \), where \( h \) is a convolution kernel, (2) is equivalent to an element-wise optimization with Fourier features.

DIP uses an encoder-decoder hourglass architecture. It consists of convolutional layers with strides for down-sampling in the encoder; convolutional layers with simple bi-linear or nearest-neighbors up-sampling in the decoder; and possibly skip-connection between corresponding scales in the encoder and decoder. For PIP we adopt the same multi-scale hourglass model, but replace each convolution with \( k \times k \) kernel and \( s \times s \) strides with a pixel-level MLP followed by nearest-neighbor downsampling (simply implemented as a convolution with 1 \(\times\) 1 kernel and \( s \times s \) strides). Fig. 1 illustrates the modifications. The suggested architecture is more efficient in terms of compute and memory as it uses 1 \(\times\) 1 convolution instead of 3 \(\times\) 3. Also, we observe that while DIP is somewhat sensitive to the architecture used (changing 3 \(\times\) 3 kernels to 1 \(\times\) 1 makes it utterly fail), this is not the case for PIP. It is much more robust to changes and we use the same model for all tasks.

Finally, unlike DIP, PIP can be easily adapted to other domains. For example, when DIP is adapted to video (using 3D convolutions), additional losses have to be used to ensure temporal consistency [41]. In PIP, the model is simply adjusted by defining Fourier features for all dimensions.

**Implementation details.** Unless otherwise specified, we follow the original architecture choice of DIP for denoising and SR: 5 levels U-net with skip connections. Each level has 2 convolutional blocks with 128 channels, and the skip connections are with convolutional blocks with 4 channels (see 5 for a detailed diagram). We also follow the original DIP hyper-parameters, Adam optimizer with a learning rate of 0.01 and the same number of iterations for early-stopping. For all demonstrated applications, we used the original configurations and only replaced the 3x3 convolutional filters with 1x1 and plugged in Fourier features as input instead of noise.
4 Evaluations

In this section, we first perform a quantitative evaluation of denoising and super-resolution (SR). We also study the effect of different design choices and understand what makes PIP work from a spectral-bias perspective. We then qualitatively demonstrate the applicability of PIP to more tasks – inpainting, blind dehazing, and CLIP inversion. For more visual results, please refer to the project website. We follow the denoising and SR experiments from DIP [36] using the same set of images. For denoising: 9 colored images\(^1\) with additive Gaussian noise (\(\sigma = 25\) and \(\sigma = 10\)) and Poisson noise. For SR: the union of Set14 [43] and Set5 [1], resulting in 19 images, with \(\times 4\) and \(\times 8\) downsampling factors.

To test PIP robustness, for each task we checked 4 PIP variants: with CNN (3 × 3 kernels) vs. MLP (1 × 1 kernels) architecture and fixed vs. learned frequencies. Table 1 summarizes the results. Observe that fixed frequencies perform slightly better for denoising while learned frequencies perform slightly better for SR. PIP exhibits robustness to the architecture used as CNN and MLP have comparable performance, while DIP relies on the CNN structure and fails with MLP.

Figures 2 and 3 show examples of the denoising and SR results, respectively, for both PIP and DIP. We observe a similar quality level overall, with DIP tending a little bit more to over-smoothing artifacts, and PIP to localized artifacts. We also observed that the performance on different images by DIP and PIP is highly correlated (Coefficient of determination \(R^2 = 0.99\)), suggesting CNN and PE have a similar ‘prior’ effect. (see Figure 16 in the appendix for more details).

To further demonstrate PIP’s superiority over recent proposed implicit models for images, we perform a quantitative evaluation as described above. Table 2 summarized our results and show that PIP is better or comparable to other methods in all reconstruction tasks.

**U-Net vs MLP.** DIP employs a U-Net architecture. It has an encoder-decoder structure, where each encoder block uses strides to downsample the image and each decoder block upsamples the image by nearest or bilinear interpolation. Parts of the original resolution signal are passed through skip connections. These down- and up-sampling operations have the effect of a low-pass filter on the parts of the signal passed through them. This can be treated as part of the image prior of the model. To test this effect we compare 1 × 1 U-Net vs. a vanilla simple MLP applied per pixel in Table 3. It shows that for denoising, a simple MLP, commonly used in implicit models, is enough and performs just as well as the U-Net that operates in a multi-scale scheme. For SR, however, U-Net is stronger than the simple MLP. This is probably due to the fact that for this task non-local information is important for retrieving lost high frequencies. Overall, PIP, unlike DIP which requires convolutions and the U-Net structure, exhibits strong robustness to the model used. Throughout this paper, for the sake of simplicity, we use the 1 × 1 U-Net architecture for all experiments.

**Activations** DIP was originally designed to work with leaky-ReLU activations. Current improvements in the field of implicit neural representations suggest replacing the vanilla relu with activations that are always differentiable such as Sine activations [34] or Gaussian activations [31]. In table 4, we incorporate those recent proposed activations into PIP U-Net architecture to measure the effect on reconstruction performance and show that in PIP for reconstruction tasks the original activation (LeakyReLU) works best compared to newer proposed activation functions.

**Effect of the Fourier Features (FF) frequency range.** The range of input frequencies allows controlling the generated image. When \(f_{\text{max}}\) in FF is too low, we generate a blurry image. When \(f_{\text{max}}\) is high we can fit high-frequency noise. However, thanks to the spectral-bias effect, early-stopping prevents fitting the noise even at high \(f_{\text{max}}\).

Figure 7 demonstrates the effect of \(f_{\text{max}}\). For low \(f_{\text{max}} = 2\) we get blurry images. We find that \(2^8 \leq f_{\text{max}} \leq 2^{10}\) provides a good balance and generates good images. But even if we use the very high \(f_{\text{max}} = 2^{16}\), thanks to the early stopping we still do not fit the noise despite the model capability. We also show the PSNR performance as a function of \(f_{\text{max}}\). The performance improves till \(f_{\text{max}} = 2^8\). Then at higher frequencies the performance varies a little bit but with no clear trend of performance degradation.

We additionally compared the frequency hyperparameter effect vs. SIREN which uses a meshgrid and sine activation. SIREN uses a frequency hyperparameter \(\omega_0\) to control the bandwidth of the output. In Figure 4 we show that SIREN is highly sensitive to the choice of \(\omega_0\) while PIP is less sensitive.

A special case of limited frequency range is using a “meshgrid” input, i.e. two input channels containing the horizontal and vertical coordinates. In FF, the encoding with the lowest frequency is sin(\(x\)), which simply applies a monotonic mapping to the coordinates and hence produces a very similar representation. Note that DIP used meshgrid as input for the

\(^1\)https://webpages.tuni.fi/foi/GCF-BM3D/index.html#ref_results
Figure 6: MLP-Denoising over the "standard dataset" with FF (PIP), Meshgrid and Noise encoding. Noise encoding fails to generate a valid mapping between the input code and the target image. The meshgrid provides a much better representation but the lack of high frequencies generates an over-smoothed image. PIP lets the model produce a suitable range of frequencies such that it denoises the image well.

Table 5: Early stopping PSNR difference in dB (↓) from the optimal stopping point.

|       | DIP | PIP |
|-------|-----|-----|
| EMV   | 1.22| 0.82|
| WMV   | 0.78| 0.75|

Table 6: Computation efficiency. (↓)

|       | Arch.| [Params][M] | FLOPs[G] | Layers |
|-------|-----|-------------|----------|--------|
| DIP   | 2.22| 65.84       | 26       |
| PIP   | 0.33| 12.15       | 26       |

Early stopping. A well known issues for DIP (and for PIP as well) is the problem of determining the number of iterations before hand. In our experiments, we follow the original DIP method and use a fixed number of iterations for early stopping (ES). However, there are methods for automatically choosing the stopping point. We tested EMV and WMV from [38] and found they work better for PIP compared to DIP. See quantitative results in Tab. 5.

Computation efficiency. We compared the original DIP’s CNN vs. our MLP in terms of parameter count, FLOPs and layer count. Despite mostly having a similar performance, thanks to the removal of all spatial filters, the number of parameters and FLOPs decreased by a factor of ~ 7 and ~ 5 with the same amount of layers; see Tab. 6.

4.1 Spectral Bias

Following [33], which analyzed DIP and showed that the model learns first the low frequencies and then the higher frequencies, we perform a similar evaluation for PIP. To test this, we fit a PIP model to a synthetic image comprised of 3 sinusoidal signals with frequencies $2\pi \cdot 4, 2\pi \cdot 8, 2\pi \cdot 16$, while the input frequency range is limited to $f_{\text{max}} = 2\pi \cdot 8$. Figure 8 presents the synthetic image as well as the images generated during the optimization process. We also calculate the Fourier transform of the generated images and observe the rate at which the GT delta functions corresponding to the original sinusoidal signals are recovered. Clearly, the generated image fits the sinusoids according to their frequency order, from small to large.

We further quantify in Figure 9 the convergence during the optimization by measuring the absolute difference between the GT and generated image Fourier transform amplitude for the 3 frequencies. Clearly, lower frequencies converge faster.

4.2 Additional Applications

We further demonstrate that PIP can serve as a drop-in replacement in other applications DIP is used for. Inpainting. We follow the examples demonstrated for DIP. In both cases, we used the learned-FF configuration and increased the number of iterations to 8k instead of 6k. Examples of our inpainting results are presented in Figs. 10 and 19. The architecture used in DIP for hole inpainting is different than the architecture used in denoising and SR in its depth (6 levels instead of 5), in its building blocks (no skip connections and bigger spatial kernels), in its embedding depth at each layer, and in the input depth (changed from 32 to 1). To make minimal adaptations to our PIP network we simply added another layer to match the same number of levels as DIP, we keep the rest of our setup exactly as described in the paper. Since PIP is less sensitive to the architecture choice, it achieves similar or better results compared to DIP with no need for re-designing the architecture for this task.

Double-PIP - Dehazing. Image dehazing is the problem of extracting a haze-free image from a hazy image. Formally, the image acquisition is modeled...
Figure 7: **FF bandwidth.** We compare image denoising results over the "standard dataset" for various $f_{\text{max}}$. As we increase $f_{\text{max}}$ the reconstruction is sharper until it reaches a plateau around $f_{\text{max}} = 2^8$. Using a very large frequency range, $f_{\text{max}} = 2^{16}$, does not drastically affect results due to the role of early-stopping in preventing noise overfitting. On the right: PSNR as a function of $f_{\text{max}}$.

Figure 8: **Spectral Bias.** We fit a PIP model to a synthetic image with 3 vertical frequencies (right column). We show the generated images as the training progresses (first row) and the Fourier transforms over a vertical line in the image (second row). We find that the model is biased towards first fitting the lower frequencies (lowest frequency at $\sim 40$ iterations; middle at $\sim 80$; highest at $\sim 200$). This explains why when the model is early-stopped the image’s lower frequencies are fitted while the high-frequency noise is removed. The highest frequency in the synthetic image is larger than $f_{\text{max}}$ in the model inputs, yet, the model is able to fit it too.

Figure 9: **Spectral Bias.** We measure the absolute difference between the predicted frequencies’ amplitude vs. the GT amplitudes. Lower frequencies are fitted first during the optimization.

as $I(x) = J(x)t(x) + A(1-t(x))$ where I is a hazy image, J is a haze-free image, A is the airlight model and $t(x) = e^{-\beta d(x)}$. In Double-DIP [9], three DIP models are trained together to generate $J(x)$ and $t(x)$ ($A$ can be derived from other algorithms or learned) with respect to the above model. Again we evaluate PIP as a drop-in replacement. We switched all noise inputs to FF and the original CNNs to our MLPs. Figs. 11 and 20 show that PIP may replace DIP also here.

**CLIP inversion.** Similarly to the original DIP inversion of AlexNet/VGG (generating images from class labels), recently DIP has been used for generating images from textual descriptions using the CLIP model [8]. DIP serves as an image generator conditioned on input text, such that the generated image embedding is close to the text embedding. Figs. 12, 13 and 17 show examples of image generation by inverting CLIP using both DIP and PIP.

### 4.3 Extension to Video

We tested an extension of PIP to video (3D-PIP) denoising and SR. To that end, we extend our positional encoding from 2D to 3D and encode the temporal domain as well as the spatial domain. Instead of stacking two spatial PEs (one for each coordinate), we
Figure 10: **Image Inpainting.** A comparison for large hole inpainting. PIP-CNN outperforms visually in both images, where PIP-MLP shows comparable results to DIP without any spatial kernels.

Figure 11: **Blind image dehazing.** We compare “Double-PIP” to “Double-DIP”, where both methods produce comparable visual results.

Figure 12: **CLIP inversion.** Generated images for the description: ‘A moody painting of a lonely duckling’.

We compare 3D-PIP to 2D-PIP (frame-by-frame) to various image prior based methods such as 2D-DIP (frame-by-frame), 3D-DIP [4] and ”Deep Video Prior” [18], as well as implicit model based models such as SIREN [34] and Gaussian activations [31]. We trained all video models for 5K iterations on all frames. We tested them on 10 videos from the DAVIS dataset [29]. Table 7 summarizes the results for all image prior based methods and Table 8 summarizes the results for all implicit models. In addition to PSNR, we also report SSIM-3D [42] to measure temporal consistency. For visual results, please refer to the project website. 2D-DIP and 2D-PIP have comparable results. 3D-DIP and DVP have better temporal consistency compared to 2D models but lack reconstruction quality. Our 3D-PIP outperforms all other models both in terms of temporal consistency and PSNR (+3dB for denoising, +0.5dB for SR). 3D-PIP achieves that with \( \times 6 \) fewer parameters compared to 3D-DIP. 3D-PIP also outperforms all INR methods by at least 2-3dB in PSNR for all restoration tasks and achieves better temporal consistency.

PIP’s high reconstruction performance can be at-
Figure 13: **CLIP inversion.** Generated images for the description: ‘An oil painting of an elephant sleeping in the desert’.

Figure 14: **Video denoising.** We compare 3D-PIP to 3D-DIP (3D conv.). 3D-PIP generates better images (e.g. the grass area) and achieves better temporal consistency (full videos in the project page).

Table 7: Image Prior Video-Denoising and SR: PSNR/3D-SSIM. 3D-PIP outperforms 2D-DIP (frame-by-frame) and 3D-DIP both in terms of pixel reconstruction (PSNR) and temporal consistency (3D-SSIM).

| Method   | $\sigma$ = 25 | $\sigma$ = 10 | Poisson | SR $x4$ |
|----------|---------------|---------------|---------|---------|
| 2D-DIP   | 26.35/0.80    | 26.63/0.81    | 26.72/0.81 | 25.54/0.81 |
| 3D-DIP   | 24.92/0.82    | 25.15/0.83    | 25.27/0.83 | 22.91/0.72 |
| DVP [18] | 25.88/0.83    | 26.01/0.83    | 25.75/0.83 | 24.29/0.79 |
| 2D-PIP   | 26.17/0.80    | 26.61/0.81    | 26.67/0.81 | 25.45/0.78 |
| 3D-PIP   | 29.29/0.9     | 30.06/0.91    | 29.97/0.91 | 26.04/0.83 |

Table 8: Implicit models Video-Denoising and SR: PSNR/3D-SSIM. 3D-PIP outperforms SIREN and Gaussian Activations both in terms of pixel reconstruction (PSNR ↑) and temporal consistency (3D-SSIM ↑).

| Method    | $\sigma$ = 25 | $\sigma$ = 10 | Poisson | SR $x4$ |
|-----------|---------------|---------------|---------|---------|
| SIREN [34]| 26.55/0.72    | 28.43/0.79    | 28.35/0.79 | 23.33/0.57 |
| Gauss. Act. [31]| 23.85/0.74 | 24.10/0.74 | 24.04/0.74 | 22.86/0.71 |
| 3D-PIP    | 29.29/0.9     | 30.06/0.91    | 29.97/0.91 | 26.04/0.83 |

5 Conclusions

In this paper we have shed a new light on DIP, treating it as an implicit model mapping shifted noise-blocks to RGB values. We have shown that Fourier features that are popular in implicit models can be used as a replacement for the noise inputs in DIP. Furthermore, when using PE, the convolutional layers can be replaced with MLPs, as commonly done in implicit models. We showed that the PE’s image prior is very similar to the CNN’s image prior, both in terms of quantitative performance (PSNR) and the visual appearance. Like DIP, we showed that PIP first fits the low-frequencies, providing an explanation on how it operates.

One limitation of PIP is that it uses the same amount of frequencies for all images. An adaptation to the frequencies in a given target image may improve the reconstruction significantly in some cases [14, 22]. In future work, we may explore selecting the input frequencies based on the input image content.
e.g. based on BACON [22]. The frequencies can also be spatially adapted, i.e., different sets of frequencies may be employed for different regions in the image as done in SAPE [14]. For example, in the baby image in Fig. 3 it would be beneficial to use lower frequencies for the face and higher for the hat. Smartly choosing the frequencies may also help to mitigate another limitation, e.g., the need for early stopping, which is also required for DIP.

Finally, some insights from DIP and PIP might help improve implicit models, e.g., using multi-scale processing in a U-Net architecture. The PE image prior that we observed might explain the ability of NeRF to interpolate between sparse sets of input views. It can also explain NeRF’s ability to learn from noisy inputs, e.g. as exhibited by RAW-NeRF [27]. Our results hint that this ability of NeRF is not just due to averaging samples from different views but also due to the impact of PE that generates well-behaved images. Finally, we demonstrated that PIP can be used for temporal signals (video denoising), where DIP based methods and INR models struggles.

Acknowledgements

This work was supported by the European research council under Grant ERC-StG 757497. This work was supported by Tel Aviv University Center for AI and Data Science (TAD).

References

[1] Marco Bevilacqua, Aline Roumy, Christine Guille- mot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012.
[2] Oscar Bustos-Brinez, Joseph Gallego-Mejia, and Fabio A. Gonzalez. Ad-dmke: Anomaly detection through density matrices and fourier features, 2022.
[3] Prithvijit Chakrabarty and Subhransu Maji. The spectral bias of the deep image prior. CoRR, abs/1912.08905, 2019.
[4] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 8628–8638, 2021.
[5] Yun-Chun Chen, Chen Gao, Esther Robb, and Jia- Bin Huang. Nas-dip: Learning deep image prior with neural architecture search. In European Conference on Computer Vision, pages 442–459. Springer, 2020.
[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.
[7] Rita Fermanian, Miakel Le Pendu, and Christine Guillemot. Regularizing the Deep Image Prior with a Learned Denoiser for Linear Inverse Problems. In MMSP 2021 - IEEE 23rd International Workshop on Multimedia Signal Processing, pages 1–6, Tampere, Finland, Oct. 2021. IEEE.
[8] Paul Fishwick. Clip inversion with dip. https://github.com/metaphorz/deep-image-prior-hqskipnet, 2021.
[9] Yosef Gandelsman, Assaf Shocher, and Michal Irani. " double-dip": Unsupervised image decomposition via coupled deep-image-priors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11026–11035, 2019.
[10] Kuang Gong, Ciprian Catana, Jinyi Qi, and Quanzheng Li. Pet image reconstruction using deep image prior. IEEE Transactions on Medical Imaging, 38(7):1655–1665, 2019.
[11] Kuang Gong, Ciprian Catana, Jinyi Qi, and Quanzheng Li. Direct reconstruction of linear parametric images from dynamic pet using nonlocal deep image prior. IEEE Transactions on Medical Imaging, 41(3):680–689, 2022.
[12] Rana Hanocka, Gal Metzer, Raja Giryes, and Daniel Cohen-Or. Point2mesh: A self-prior for deformable meshes. ACM Trans. Graph., 39(4), 2020.
[13] Fumio Hashimoto, Kibo Ote, and Yuya Onishi. PET image reconstruction incorporating deep image prior and a forward projection model. IEEE Transactions on Radiation and Plasma Medical Sciences, 6(8):841–846, nov 2022.
[14] 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.
[15] Xin Jin, Li Zhang, Chaowei Shan, Xin Li, and Zhibo Chen. Dual prior learning for blind and blended image restoration. IEEE Transactions on Image Processing, 31:1042–1056, 2022.
[16] Chaewon Kim, Jaeho Lee, and Jinwoo Shin. Zero-shot blind image denoising via implicit neural representations. arXiv preprint arXiv:2204.02405, 2022.
[17] Edwin Kurniawan, Yunjin Park, and Sukho Lee. Noise-resistant demosaicing with deep image prior network and random rgbw color filter array. Sensors, 22(5):1767, 2022.
[18] Chenyang Lei, Yazhou Xing, and Qifeng Chen. Blind video temporal consistency via deep video prior. Advances in Neural Information Processing Systems, 33:1083–1093, 2020.
[19] Daniel LeJeune, Lorenzo Luzi, Ali Siahkoohi, Sina Alemohammad, Vishwanath Saragadam, Hossein Babaei, Naiming Liu, Zichao Wang, and Richard G Baraniuk. Titan: Bringing the deep image prior to implicit representations. arXiv preprint arXiv:2211.00219, 2022. 3, 4

[20] Alexander Li and Deepak Pathak. Functional regularization for reinforcement learning via learned fourier features. Advances in Neural Information Processing Systems, 34:19046–19055, 2021. 3

[21] Yang Li, Si Si, Gang Li, Cho-Jui Hsieh, and Samy Bengio. Learnable fourier features for multi-dimensional spatial positional encoding. Advances in Neural Information Processing Systems, 34:15816–15829, 2021. 5

[22] David B. Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. Bacon: Band-limited coordinate networks for multiscale scene representation. In CVPR, 2022. 3, 10, 11

[23] Erika Lu, Forrester Cole, Tali Dekel, Andrew Zisserman, William T Freeman, and Michael Rubinstein. Ommimatte: Associating objects and their effects in video. In CVPR, 2021. 4, 9

[24] Gary Mataev, Michael Elad, and Peyman Milanfar. Deepred: Deep image prior powered by red. ArXiv, abs/1903.10176, 2019. 3

[25] Michael Mathieu, Mikael Henaff, and Yann LeCun. Fast training of convolutional networks through ffts. arXiv preprint arXiv:1304.0038, 2013. 14

[26] B Mildenhall. Nerf: Representing scenes as neural radiance fields for view synthesis. In European conference on computer vision, 2020. 1

[27] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P Srinivasan, and Jonathan T Barron. Nerf in the dark: High dynamic range view synthesis from noisy raw images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16190–16199, 2022. 3, 11

[28] Himashi Peiris, Munawar Hayat, Zhaolin Chen, Gary Egan, and Mehrtaash Harandi. A robust volumetric transformer for accurate 3d tumor segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 162–172. Springer, 2022. 3

[29] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeléiz, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 daviscv challenge on video object segmentation. arXiv preprint arXiv:1704.00675, 2017. 9

[30] Nasim Rahaman, Aристide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In International Conference on Machine Learning, volume 97, pages 5301–5310, 2019. 2

[31] Sameera Ramasinghe and Simon Lucey. Beyond periodicity: towards a unifying framework for activations in coordinate-mlps. In Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII, pages 142–158. Springer, 2022. 3, 6, 9, 10

[32] Dongwei Ren, Kai Zhang, Qiong Long, Qinghua Hu, and Wangmeng Zuo. Neural blind deconvolution using deep priors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3341–3350, 2020. 3

[33] Zenglin Shi, Pascal Mettes, Su hranu Maji, and Cees G M Snoek. On measuring and controlling the spectral bias of the deep image prior. International Journal of Computer Vision, 2022. 1, 3, 7

[34] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In arXiv, 2020. 3, 6, 9, 10

[35] Matthew Tancik, Pratul P Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. NeurIPS, 2020. 3

[36] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 9446–9454, 2018. 1, 3, 4, 6, 10, 15

[37] Ashish Vaswani, Noam Shazeer, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ulasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017. 2

[38] Hengkang Wang, Taihui Li, Zhong Zhuang, Tiancong Chen, Hengyue Liang, and Ju Sun. Early stopping for deep image prior. Transactions on Machine Learning Research, 2023. 3, 7

[39] Rui Xu, Xintao Wang, Kai Chen, Bolei Zhou, and Chen Change Loy. Positional encoding as spatial inductive bias in gans, 2020. 3

[40] Tatsuya Yokota, Kazuya Kawai, Muneyuki Sakata, Yuichi Kimura, and Hidekata Hontani. Dynamic pet image reconstruction using nonnegative matrix factorization incorporated with deep image prior. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2019. 3

[41] Jaejun Yoo, Kyong Hwan Jin, Harshit Gupta, Jerome Yerly, Matthias Stuber, and Michael Unser. Time-dependent deep image prior for dynamic mri. IEEE Transactions on Medical Imaging, 2021. 4, 5

[42] Kai Zeng and Zhou Wang. 3d-ssim for video quality assessment. In 2012 19th IEEE International Conference on Image Processing, pages 621–624, 2012. 9

[43] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In International conference on curves and surfaces, pages 711–730. Springer, 2010. 6
[44] Haotian Zhang, Long Mai, Ning Xu, Zhaowen Wang, John Collomosse, and Hailin Jin. An internal learning approach to video inpainting. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2019. 1, 3, 10

[45] Jenny Zukerman, Tom Tirer, and Raja Giryes. Bp-dip: A backprojection based deep image prior. In 2020 28th European Signal Processing Conference (EUSIPCO), pages 675–679. IEEE, 2021. 3
Appendix

A Proof of Proposition 1

Proposition 1. For \( E \) being the \( \ell_2 \) loss and \( f(z) = h \ast z \), where \( h \) is a convolution kernel of size \( r \), problem (2) is equivalent to an element-wise optimization with Fourier features.

Proof. We are examining the case of 1D denoising using a single linear layer without any activation. Input is as described in DIP, A uniform noise from \( n \sim U[0, \frac{1}{m}] \).

The following optimization is performed:

\[
\| h \ast n - y \|_2^2, 
\]

where \( h \) is our spatial kernel, \( n \) is the input noise and \( y \) is the denoised image. Following [25], we can formulate the spatial convolutions into an element-wise product in the frequency domain:

\[
\| F^{-1} \{ F(h) \cdot F(n) \} - y \|_2^2. 
\]

We will turn to focus on the left term. Denote by \( F_N \) the \( N \times N \) Discrete Fourier Transform (DFT) matrix:

\[
F_N = \begin{pmatrix}
1 & 1 & 1 & \ldots & 1 \\
1 & \omega_N^1 & \omega_N^2 & \ldots & \omega_N^{N-1} \\
1 & \omega_N^2 & \omega_N^4 & \ldots & \omega_N^{2(N-1)} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & \omega_N^{N-1} & \omega_N^{2(N-1)} & \ldots & \omega_N^{(N-1)^2}
\end{pmatrix},
\]

where,

\[
\omega_N^{-1} = e^{-i2\pi/N} = e^{i2\pi/N} = \overline{\omega_N}.
\]

Thus, we may write the Fourier transformation in a matrix form as follows, \( \tilde{n} = F(n) =
\]

\[
\begin{pmatrix}
1 & 1 & 1 & \ldots & 1 \\
1 & \omega_N & \omega_N^2 & \ldots & \omega_N^{N-1} \\
1 & \omega_N^2 & \omega_N^4 & \ldots & \omega_N^{2(N-1)} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & \omega_N^{N-1} & \omega_N^{2(N-1)} & \ldots & \omega_N^{(N-1)^2}
\end{pmatrix}
\begin{pmatrix}
n_0 \\
n_1 \\
n_2 \\
\vdots \\
n_{N-1}
\end{pmatrix}.
\]

and similarly, \( \tilde{h} = F(h) =
\]

\[
\begin{pmatrix}
1 & 1 & 1 & \ldots & 1 \\
1 & \omega_N^1 & \omega_N^2 & \ldots & \omega_N^{N-1} \\
1 & \omega_N^2 & \omega_N^4 & \ldots & \omega_N^{2(N-1)} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & \omega_N^{N-1} & \omega_N^{2(N-1)} & \ldots & \omega_N^{(N-1)^2}
\end{pmatrix}
\begin{pmatrix}
h_0 \\
h_1 \\
h_2 \\
\vdots \\
h_{N-1}
\end{pmatrix}.
\]

Notice that since \( F_N \) is unitary and \( n \) is initialized as i.i.d. random Gaussian with zero mean, then \( \tilde{n} \) is also i.i.d. random Gaussian with zero mean. Note also that \( \tilde{h} \) is basically a (random) linear combination of Fourier features. Now, note that \( \hat{y} \triangleq F^{-1}(\tilde{n} \cdot \tilde{h}) =
\]

\[
\frac{1}{N}
\begin{pmatrix}
1 & 1 & 1 & \ldots & 1 \\
1 & \overline{\omega_N^1} & \overline{\omega_N^2} & \ldots & \overline{\omega_N^{N-1}} \\
1 & \overline{\omega_N^2} & \overline{\omega_N^4} & \ldots & \overline{\omega_N^{2(N-1)}} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & \overline{\omega_N^{N-1}} & \overline{\omega_N^{2(N-1)}} & \ldots & \overline{\omega_N^{(N-1)^2}}
\end{pmatrix}
\begin{pmatrix}
\tilde{h}_0 \cdot \tilde{n}_0 \\
\tilde{h}_1 \cdot \tilde{n}_1 \\
\tilde{h}_2 \cdot \tilde{n}_2 \\
\vdots \\
\tilde{h}_{N-1} \cdot \tilde{n}_{N-1}
\end{pmatrix}
\]

Focusing at the \( k^{th} \) row, we have that

\[
\hat{y}(k) = \frac{1}{N} \sum_{l=0}^{N-1} \omega_N^{kl}(\tilde{h}_l \cdot \tilde{n}_l). 
\]

Since \( n \) and \( h \) are real signals, both are symmetric in the frequency domain. (namely \( \tilde{n}[k] = \tilde{n}[N-k] \)).

and \( \omega_N^{kl} \) is periodic \( N \), we can write: (*up to \( N \) odd/even)

\[
\hat{y}(k) = \frac{2}{N}(\tilde{h}_0 \cdot \tilde{n}_0) + \frac{1}{N} \sum_{l=1}^{N/2-1} (\omega_N^{kl} + \omega_N^{-kl})(\tilde{h}_l \cdot \tilde{n}_l)
\]

Which is equivalent to taking only the real part in the spatial domain (of a signal with a length of \( \frac{N}{2} \)).

Using the relations between complex exponents and cosine we get the following: \( \frac{2\pi}{N} \cdot m = f_m \) we get that we can re-write the \( k^{th} \) row as follows:

\[
\hat{y}(k) = \frac{2}{N} \sum_{l=1}^{N/2-1} \cos(f_l \cdot k) \cdot (\tilde{h}_l \cdot \tilde{n}_l)
\]

From the equation, one can see that the left term in Equation (4) is equivalent to a sum of cosines that element-wise multiplied by \( \tilde{h}_l \) is the term that is being optimized (through the entries of \( h \)). In PIP, we directly optimize \( \hat{h}_l \) and treats its entries independently.

\[\square\]
Figure 15: **PIP input encoding.** For image restoration we used a 2D encoding, concatenating vertical and horizontal FF-PE. When PIP is extended to video, we concatenate a temporal FF encoding based on all the frames (normalized from 0 (first frame) to 1 (last frame)).

Figure 16: DIP and PIP correlation on Image denoising (\(\sigma = 25\)), the measured "the coefficient of determination" \((R^2)\) is 0.99 suggesting PIP and DIP has similar image prior.

Figure 17: **CLIP inversion.** Generated images for the description: ‘A dark oil painting of a sheep in the middle of the sea’.

Figure 18: **Image denoising \((\sigma = 25)\) with meshgrid.** Comparison of PIP to DIP with meshgrid encoding. Note that for DIP with meshgrid as input the results are over-smoothed and the model is unable to recover details.

Figure 19: **Image inpainting.** DIP vs. PIP comparison.
Figure 20: **Blind image dehazing.** Double-DIP and Double-PIP qualitative comparison.