This CVPR 2021 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Positional Encoding as Spatial Inductive Bias in GANs

Rui Xu[1] Xintao Wang[3] Kai Chen[4,5] Bolei Zhou[1] Chen Change Loy[2]
[1] CUHK-SenseTime Joint Lab, The Chinese University of Hong Kong [2] S-Lab, Nanyang Technological University
[3] Applied Research Center, Tencent PCG [4] SenseTime Research [5] Shanghai AI Laboratory
{xr018, bzhou}@ie.cuhk.edu.hk xintao.wang@outlook.com
chenkai@sensetime.com ccloy@ntu.edu.sg

Abstract

SinGAN shows impressive capability in learning internal patch distribution despite its limited effective receptive field. We are interested in knowing how such a translation-invariant convolutional generator could capture the global structure with just a spatially i.i.d. input. In this work, taking SinGAN and StyleGAN2 as examples, we show that such capability, to a large extent, is brought by the implicit positional encoding when using zero padding in the generators. Such positional encoding is indispensable for generating images with high fidelity. The same phenomenon is observed in other generative architectures such as DCGAN and PGGAN. We further show that zero padding leads to an unbalanced spatial bias with a vague relation between locations. To offer a better spatial inductive bias, we investigate alternative positional encodings and analyze their effects. Based on a more flexible positional encoding explicitly, we propose a new multi-scale training strategy and demonstrate its effectiveness in the state-of-the-art unconditional generator StyleGAN2. Besides, the explicit spatial inductive bias substantially improves SinGAN for more versatile image manipulation.[1]

1. Introduction

SinGAN [42] and StyleGAN [21, 22] are among the few representative Generative Adversarial Networks (GANs) that show impressive image generative capability. Both of their generators are based on a fully translation-invariant convolutional network. One would expect that in an unconditional setting with a spatially *i.i.d.* input, the translation invariance property should result in position-agnostic outputs like Fig. 1(b). Nonetheless, the results of SinGAN shows surprisingly structured results like Fig. 1(a).

We carefully study this phenomenon and find that it is the zero padding that causes a location-aware bias in the distribution of feature maps. Such a spatial bias gradually spreads from the border to the center of feature maps through the stacked convolutional layers in the generator. One can regard this spatial bias as an implicit positional encoding, which contributes to the high fidelity of images generated by SinGAN and StyleGAN. Interestingly, we also observe this phenomenon in other unconditional generative architectures such as DCGAN [36] and PGGAN [20].

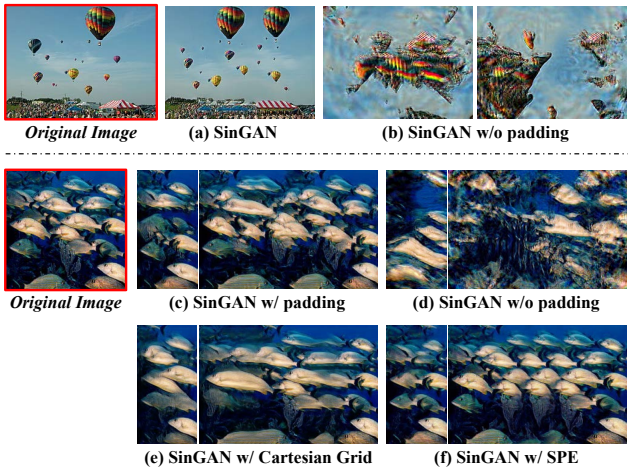


Figure 1: Images sampled from the internal patch distribution learned by SinGAN. Above the dotted line, we present sampled balloons with standard SinGAN and padding-free SinGAN. A more challenging case of generating a school of fish is shown below the dotted line. (c)-(f) show the effects of different positional encodings that we explore on SinGAN.

Our observation reveals the importance of introducing positional encoding in generative models. The original intention of zero padding is to maintain the spatial size of feature maps. It is not specially designed to offer the required spatial inductive bias. In particular, we find that the bias caused by zero padding is unbalanced over the image space. Since paddings are introduced at image borders and corners, the positional encoding at those locations is structured. In contrast, the spatial encoding in the center region is highly unstructured due to the gradually diminishing effects of zero padding from borders to the center. The shortcoming of such bias can be observed from Fig. 1 where we use SinGAN

[1]Project page: https://nbei.github.io/gan-pos-encoding.html

to synthesize an image of a school of fish. In this example, SinGAN generates relatively more structured output at the borders but inferior results at the center of the image.

The aforementioned example suggests the shortcoming of padding in serving the role of positional encoding. The desired positional encoding should keep a consistent spatial structure and be invariant to scale transformation. In this study, we investigate two alternatives for explicit positional encoding, *i.e*., the normalized Cartesian spatial grid [19] and 2D sinusoidal positional encoding [9, 32, 34, 47], which both guarantee a balanced spatial inductive bias over the whole image space. We show that these explicit positional encodings allow a convolutional generator to generate images that exhibit a more stable structure and more reasonable patch reoccurrence given an arbitrary scale in the generation, as shown in Fig. 1(e) and Fig. 1(f).

With the more flexible explicit positional encodings, we can redesign convolutional generative models to synthesize images at multiple scales even just using a single model. Achieving this functionality is challenging with existing models. One will typically need to train different generators with different upsampling blocks. We show that multi-scale generation with a single fully convolutional generator is possible using our newly proposed multi-scale training strategy based on explicit positional encodings. We call it Multi-Scale training with PositIon Encodings (MS-PIE). We demonstrate its effectiveness in the state-of-the-art unconditional generator StyleGAN2. With MS-PIE, a single StyleGAN2 that is designed for 256×256 image generation yields compelling generation quality at multiple scales up to 512×512 or even 1024×1024 pixels, despite that it only contains limited upsampling blocks in its architecture.

We summarize the contributions of this study as follows: (1) we reveal the phenomenon where zero padding unintentionally introduces implicit (but useful) positional encoding in existing convolutional generators. We study this phenomenon through detailed theoretical and empirical analyses. While the influence of padding to translation-invariant convolution has been discussed in recent works [2, 18, 23], these studies focus on image classification and detection. Our research is the first study that investigates the impacts of such spatial bias on image generation. (2) We further investigate and present two explicit positional encodings as two new spatial inductive bias in generators, which can substantially improve the versatility and robustness of SinGAN. (3) We propose a new multi-scale training strategy to achieve high-quality multi-scale synthesis with a single StyleGAN2 that is originally designed for 256×256 generation.

2. Related Work

Padding Effects. Some recent studies [2, 18, 23] discover an intriguing phenomenon in which the widely used zero padding would offer spatial information (an unintended design) in convolutional networks for image classification [8, 14] and detection [12, 39]. Islam *et al*. [18] find padding implicitly injects positional information in ResNet [14] and VGG [45], verified with an auxiliary positional encoding module. The experiments in [23] further show that the effects of padding vary among different architectures [4, 16]. Alsallakh *et al*. [2] observe the similar phenomenon and find that such spatial bias would cause blind spots for detectors [28]. In this work, we show both theoretically and empirically how zero padding accidentally encodes spatial information for convolutional generators. We find that spatial bias, unlike high-level visual tasks, is actually necessary for generators to work well. We further discuss better choices of spatial inductive bias.

Sinusoidal Positional Encoding. Sinusoidal positional encoding (SPE) is widely used in natural language processing (NLP) [6, 9, 47] and 3D vision [32, 34, 46]. In the transformer architecture [47], the sequence model relies on SPE to indicate the time step. SPE provides a stable and reasonable positional encoding for dealing with natural language because the transformation between different time steps in SPE is irrelevant to the length of the input sentence. To avoid the spectral bias [37] in the fully connected networks, Martin *et al*. [32] transfers the input features from the low-frequency domain to the high-frequency domain [46]. Different from the aforementioned studies, we focus on adopting sinusoidal positional encoding in 2D convolutional generators to obtain more effective spatial inductive bias.

Cartesian Grid. Cartesian grid has been introduced in spatial transformer networks [19] as a standard coordinate system for 2D spatial feature space. It is widely adopted for differentiable image warping [38, 44, 49, 51, 50, 48] and aligning pixels among different spaces [10, 17, 27, 40, 41, 53]. In this study, we develop a new role of explicit positional encoding for the normalized Cartesian grid.

3. Methodology

As shown in Fig. 1, once we remove the padding in SinGAN, we observe that the translation-invariant convolutional generator collapses to position-agnostic distribution. This suggests that SinGAN relies on zero padding to capture spatial information. From the view of the stochastic process, we clarify how zero padding works as implicit positional encoding. After analyzing such implicit positional encoding, we investigate the potential of two explicit positional encodings as better spatial inductive bias in GAN's generator. Finally, we present applications on MS-PIE and SinGAN to prove the significance of spatial inductive bias to GANs.

3.1. Translation Invariance in Generative Models

To better understand the effects of zero padding, we first analyze the behavior of padding-free convolutional generators. The popular SinGAN adopts a fully convolutional

generator with a spatially *i.i.d.* noise map as input. Thus, the translation-invariant convolutional network can be regarded as a stochastic process on spatial random variables. We mainly study two basic statistical properties of the expectation ($\mathbb{E}$) and autocorrelation function (R) for the convolutional feature maps. $\mathbb{E}(y_{\vec{\mathbf{i}}})$ defines the distributional property of each location $\vec{\mathbf{i}}$ in the convolutional feature map $\{y_{\vec{\mathbf{i}}}\}$, while $R(y_{\vec{\mathbf{i}}}, y_{\vec{\mathbf{j}}})$ depicts the relationship between two spatial locations. Due to the space limitation, a detailed derivation is shown in our supplementary material.

Taking $x_k \in X_{\vec{\mathbf{i}}} \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$ as input, the expectation of the feature map after the first convolutional layer $y_{\vec{\mathbf{i}}}^{(1)}$ is:

$$\begin{aligned}\mathbb{E}(y_{\vec{\mathbf{i}}}^{(1)}) &= \sum_k w_k^{(1)} \int_{-\infty}^{+\infty} x_k p(x_k) dx_k + b^{(1)} \\ &= \sum_k w_k^{(1)} \mathbb{E}(x_k) + b^{(1)} = b^{(1)}, \end{aligned} \tag{1}$$

where $(w_k^{(1)}, b^{(1)})$ are parameters in the first convolutional layer and the subscript k indicates the k-th item in the sum of a convolutional weight (w_k) multiplying an input feature (x_k). Besides, $p(\cdot)$ represents the probability density function. As we assume $X_{\vec{\mathbf{i}}} \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$, the zero $\mathbb{E}(x_k)$ induces that the expectation of the first convolutional feature is only related to the bias parameter. After applying a commonly used LeakyReLU function (g) with negative slope (γ) [31] and the second convolutional layer, a more general formulation for the expectation $\mathbb{E}(y_{\vec{\mathbf{i}}}^{(2)})$ should be:

$$\begin{aligned}\mathbb{E}(y_{\vec{\mathbf{i}}}^{(2)}) &= \sum_k w_k^{(2)} \int_{-\infty}^{+\infty} g(y_k^{(1)}) p(y_k^{(1)}) dy_k^{(1)} + b^{(2)} \\ &= \sum_k w_k^{(2)} \cdot (\gamma \mathbb{C}_1 + \mathbb{C}_2) + b^{(2)}, \end{aligned} \tag{2}$$

where $\mathbb{C}_1, \mathbb{C}_2$ are constants from the finite piecewise-defined integration. Eq. (2) shows that the convolutional features keep a spatially identical expectation value, which is a linear combination of the convolutional weights. Furthermore, the analysis in the autocorrelation function[2] further shows that the relation of two positions in the first convolutional feature map is decoupled with the absolute position:

$$\begin{aligned} R(y_{\vec{\mathbf{i}}}, y_{\vec{\mathbf{j}}}) &= \mathbb{E}(y_{\vec{\mathbf{i}}} y_{\vec{\mathbf{j}}}) \\ &= \sum_{x_l \in X_{\vec{\mathbf{i}}} \cap X_{\vec{\mathbf{j}}}} w_{k_l} w_{t_l} \mathbb{E}(x_l^2) \\ &= R(\vec{\mathbf{i}} - \vec{\mathbf{j}}). \end{aligned} \tag{3}$$

Here, the condition $x_l \in X_{\vec{\mathbf{i}}} \cap X_{\vec{\mathbf{j}}}$ determines whether x_l belongs to the intersection region of related input features. If there is no intersection between two input features, the sum in Eq. (3) will be zero. Importantly, the intersection region $X_{\vec{\mathbf{i}}} \cap X_{\vec{\mathbf{j}}}$ is determined by the offset vertor $\vec{\mathbf{i}} - \vec{\mathbf{j}}$. Thus, the autocorrelation function of $(y_{\vec{\mathbf{i}}}, y_{\vec{\mathbf{j}}})$ is only related to $\vec{\mathbf{i}} - \vec{\mathbf{j}}$ but irrelevant to the absolute position vector $\{\vec{\mathbf{i}}, \vec{\mathbf{j}}\}$. This proves that after convolution, the features can be regarded as a spatial ***weak stationary stochastic process***.

An essential property of weak stationarity is that the absolute positional information is lost. As shown in Fig. 1(b), without any spatial bias, convolutional generators fail to capture faithful spatial structures, *e.g.*, the position of the ground and the spatial organization of related patches (balloons). However, it can still output some reasonable patches like balloon texture patterns. The reason is that the truncated $R(y_{\vec{\mathbf{i}}}, y_{\vec{\mathbf{j}}})$ models the relationship between convolutional features within the limited effective receptive field. In conclusion, the translation invariance in convolution leads to weak stationarity in features.

3.2. Padding as Spatial Inductive Bias

As discussed in Sec. 3.1, the translation-invariant convolution causes positional information loss from the convolutional features. Thus, SinGAN should have generated results without reasonable spatial structures like Fig. 1(b). However, as shown in Fig. 1(a), zero padding unintentionally enables SinGAN to capture the spatial structure of the sky, ground, and *etc*. Based on the analysis in Sec. 3.1, we will clarify this phenomenon theoretically.

From the view of the effective receptive field [30], we regard the whole convolutional network as a convolutional layer with a large kernel and move all of paddings to the input, which is illustrated in Fig. 2. Then, the linear combination of the convolutional kernel weights in Eq. (2) and Eq. (3) will be influenced by zero padding:

$$\mathbb{E}(y_{\vec{\mathbf{i}}}) = \sum_k w_k (\gamma \mathbb{C}_1 + \mathbb{C}_2) \mathbb{1}(x_k \notin Pad) + b, \tag{4}$$

$$R(y_{\vec{\mathbf{i}}}, y_{\vec{\mathbf{j}}}) = \sum_{x_l \in X_{\vec{\mathbf{i}}} \cap X_{\vec{\mathbf{j}}}} w_{k_l} w_{t_l} \mathbb{E}(x_l^2) \mathbb{1}(x_l \notin Pad), \tag{5}$$

where the indicator function $\mathbb{1}(x_i \notin Pad)$ determines if the current input belongs to the padding regions. Intuitively, when the convolution kernel meets input features containing zero padding, some convolutional weights will be multiplied by the zero value. The number of such inevitable zero terms varies as the convolution kernel slides over the feature map. Namely, the padding effects on Eq. (4) and Eq. (5) are determined by the overlap of the convolution kernel and zero padding. Therefore, zero padding implicitly injects positional information through the location-variant $\mathbb{E}(y_{\vec{\mathbf{i}}})$ and $R(y_{\vec{\mathbf{i}}}, y_{\vec{\mathbf{j}}})$. The $\mathbb{E}(y_{\vec{\mathbf{i}}})$ in Eq. (4) yields the position-aware distribution of the spatial random variables, which is a kind of implicit positional encoding. In our supplementary material, we will show that the location-variant $R(y_{\vec{\mathbf{i}}}, y_{\vec{\mathbf{j}}})$ can be applied to explain the behavior of other padding modes.

3.3. Analysis on Implicit Positional Encoding

The implicit positional encoding introduced by zero padding offers unbalanced spatial inductive bias over the

[2] We discard the bias term since the identical addition term does not influence the final conclusion. See complete derivations in the appendix.

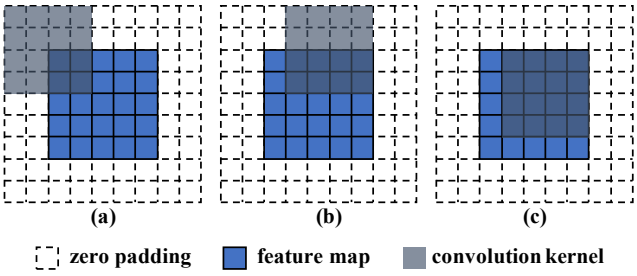


Figure 2: Illustration for the convolutional procedure with zero padding. We move the padding in each layer to the input feature and regard the whole convolutional network as a convolutional layer with a large kernel.

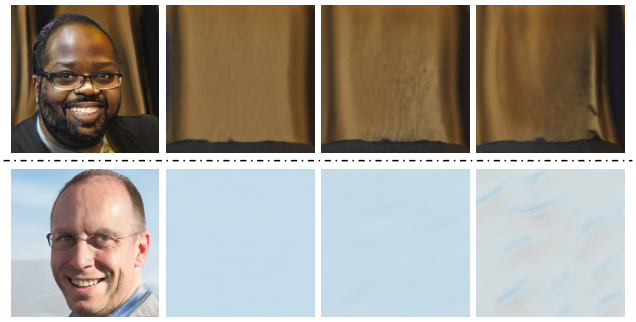


Figure 3: Images sampled from StyleGAN2 (above the dotted line) and padding-free StyleGAN2 (under the dotted line). The first column is sampled with the original learned constant input. The other three columns are sampled with different identical values (from left to right: 0, 0.5, 1) filling in the learned constant input at the start of the generator.

whole image space. As shown in Eq. (4), the top-left corner in Fig. 2 will receive a unique position encoding, because such an overlap of zero padding and the convolutional kernel must only exist at the top-left corner. However, as the convolution kernel slides away from corners and borders, the central locations (Fig. 2(c)) will be encoded with the same positional information. An important property of this implicit positional encoding is that distinct ***spatial anchors*** provide fixed yet definite spatial bias at corners. Nevertheless, the distance between two positional encodings becomes uncertain (or even zero) in the central regions. Following the definition in transformer [47], we call such distance as ***transformation between locations***, because we mainly care about how to transform from one location to another one in the positional encoding.

Such unbalanced spatial inductive bias is not ideal for image generation. Intuitively, without a clear transformation between locations, the uniqueness of the positional encoding cannot be guaranteed. Consequently, the convolutional generator fails to precisely portray the desired objects at the center of the generated image, as shown in Fig. 1(c). Furthermore, implicit spatial anchors also bring frozen structures at borders. To show this padding effect on StyleGAN2, we fill in the constant input ahead of its convolutional generator with an identical value. The identical constant input should have caused spatially consistent patterns because of the weak stationarity in convolutional features. Nevertheless, Fig. 3 shows generated images with borders of similar structure. Such frozen structures suggest the strong influence from zero padding, which causes the generator to overfit several spatial structures in the training distribution. Once the padding is removed, spatially identical color or pattern will cover the whole image, as shown in the second row of Fig 3. In addition, a shift from precision to recall [22, 26] is observed in our experiments on the padding-free StyleGAN2, suggesting that zero padding limits the diversity of a generative model.

Prior to SinGAN and StyleGAN, generators typically use a fully connected layer [11] to take the noise vector as input.[3] Followed by a reshaping layer, this operation explicitly injects spatial information to the feature map ahead of convolutional blocks [3, 5]. Our experiments show that DCGAN [36] and PGGAN [20] still rely on the implicit positional encoding to a much greater extent than we expect.

3.4. Explicit Positional Encoding for GANs

The analysis in Sec. 3.3 clarifies that implicit positional encoding cannot provide a balanced spatial inductive bias and cannot keep spatially consistent transformation between positions (Fig. 4(a)). In this section, we will discuss three explicit positional encodings and analyze the spatial inductive bias introduced by them.

Learnable Constant Input. In StyleGAN [21, 22], they adopt a $4 \times 4 \times 512$ learnable constant as the input of the convolutional generator. The learnable constant input is fixed across samples but it offers a unique 512-dimension vector as a learned positional encoding for the 4×4 input space. However, the chaotic structures in Fig. 4(b) illustrate that the spatial inductive bias defined by the learnable constant input is unclear and lacks explicit priors on image space.

Cartesian Spatial Grid. The Cartesian spatial grid (CSG), used in spatial transformer network [19], can play a role of positional encoding. To avoid large value at huge input space, the Cartesian spatial grid mentioned in this work is normalized as:

$$\vec{P}_{CSG}(i,j) = 2 \cdot [\frac{i}{H} - \frac{1}{2}, \frac{j}{W} - \frac{1}{2}], \tag{6}$$

where (i, j) represents the spatial location in the $H \times W$ input space. Thus, the corners and central points are fixed to a constant vector, *e.g.*, $[-1, -1]$ for the top-left corner and $[0, 0]$ for the central point. Such fixed and distinct reference points provide spatial anchors across the image space. Besides, the transformation between locations is spatially consistent at a single scale:

$$\vec{P}_{CSG}(i,j) = 2 \cdot [\frac{i - i'}{H}, \frac{j - j'}{W}] + \vec{P}_{CSG}(i',j'). \tag{7}$$

[3] Although, in some implementations, they use transposed convolution on 1×1 noise vector, it equals to applying a linear layer mathematically.

Table 1: Summary of spatial inductive bias defined by different positional encodings. 'Identical Transform' with 'SS' means the transformation between locations is spatially identical at a single scale. The 'MS' row shows whether the transformation is scale-invariant. 'Interp' is the traditional interpolation while 'Expand' denotes the expansion in SPE.

Spatial Inductive Bias		Pad	Learnable Constant	Cartesian Grid	SPE
Spatial Anchors		✓		✓	
Identical Transform	SS			✓	✓
	MS				✓
Resize Mode	Interp		✓	✓	✓
	Expand				✓

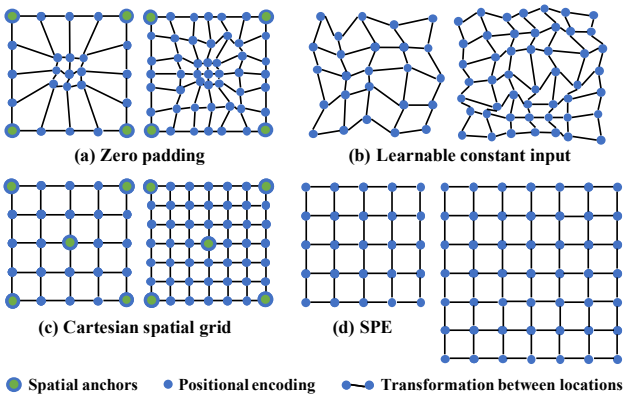


Figure 4: Illustration for the 2D spatial space defined by different positional encodings. For each encoding, we present two spatial spaces at 5×5 and 7×7 scale. (d) shows how SPE naturally expands its space with consistent transformation.

However, the transformation in Eq. (7) is related to the input scale (H, W). As shown in Fig. 4(c), a larger input scale will bring closer distances between adjacent positional encodings, despite that spatial anchors are fixed. Thus, the Cartesian grid is robust to align global structures across multiple scales (Fig. 1(e)), but the detailed structure will be interpolated similarly with the resizing mode of the Cartesian grid.

Sinusoidal Positional Encoding. Sinusoidal Positional Encoding (SPE) has been widely adopted in NLP [9, 47] and 3D vision [32, 34]. To construct 2D positional encoding, we concatenate the encodings in height and width dimensions:

$$[\underbrace{sin(\omega_0 i), cos(\omega_0 i), \cdots}_{height\ dimension}, \underbrace{sin(\omega_0 j), cos(\omega_0 j), \cdots}_{width\ dimension}], \quad (8)$$

where $\omega_k = 1/10000^{2k/d}$ and d denotes half of the total encoding dimension. The formulation in Eq. (8) guarantees that the transformation between locations is decoupled with input scales and only related to the offset positional vector:

$$\begin{bmatrix} sin(\omega_k i) \\ cos(\omega_k i) \end{bmatrix} = \begin{bmatrix} cos(\omega_k \phi) & sin(\omega_k \phi) \\ -sin(\omega_k \phi) & cos(\omega_k \phi) \end{bmatrix} \cdot \begin{bmatrix} sin(\omega_k i') \\ cos(\omega_k i') \end{bmatrix}, \quad (9)$$

where $\phi = i - i'$ is the positional offset. Thus, without spatial anchors, SPE can naturally expand its space by extending more pixels while keeping the consistent transformation between adjacent positions, as illustrated in Fig. 4(d). With such a scale-agnostic transformation, the detailed structure will not be affected when we change the input scale. Thanks to the stable spatial inductive bias, SPE shows impressive capability in constructing realistic patch recurrence (Fig. 1(f)).

Table 1 summarizes the spatial inductive bias that is contained in different positional encodings. In the following two sections, we will present two applications with various positional encodings and further discuss the significance of spatial inductive bias to convolutional generators.

3.5. Multi-scale Training with Positional Encoding

As shown in Tab. 1, an explicit positional encoding can be resized to different scales by either interpolation or expansion. Inspired by this property, we derive a new training strategy for performing multi-scale synthesis with a single fully convolutional generator. Typically, one usually fixes the input scale, like 4×4, and depends on the different number of upsampling blocks to generate multi-scale images.

Contrary to the above practice, we show that by resizing the explicit positional encoding ahead of the convolutional generator, one can generate images with compelling quality at multiple scales. We call our method as Multi-Scale training with Positional Encoding (MS-PIE). Based on 256^2 StyleGAN2, we demonstrate the effectiveness of MS-PIE and present the impacts of spatial inductive bias in the padding-based and padding-free settings. Directly adopting the original learnable constant input in StyleGAN2 causes inferior generation quality due to the lack of any explicit priors on the dynamic image space. With the standard StyleGAN2 containing zero padding, the explicit scale-invariant transformation (Eq. (9)) in SPE provides a much precise spatial inductive bias over the dynamic input space. Therefore, the generator designed for the scale of 256^2 succeeds in high-quality image synthesis at multiple scales up to 512×512 or even 1024×1024 pixels. As for the padding-free setting, we discover that the fixed spatial anchors in CSG are essential for the generator to align the global structures among different scales. It is the explicit spatial anchors that mitigate the effects of removing zero padding in each layer, which leads to superior performance in the padding-free setting.

The ultimate goal of MS-PIE is to effectively leverage different image scales for high-fidelity image generation. Intuitively, the spatial structure can be efficiently captured in low-resolution space. The spatial inductive bias guides the generators to enlarge image space according to the resizing mode of the input positional encoding. Meanwhile, thanks to the priors on dynamic image space and the mixed training of multi-scale images, the high-resolution texture space can be transferred from the low-resolution domain efficiently. Thus, in our MS-PIE, the generative model is trained on high-resolution images with fewer iterations. In each iteration, we sample the current training scale according to a given

probability where the higher probability is set for the lower scale. Then, the real image resolution and the size of the explicit input positional encoding are modified accordingly. Besides, to keep the input dimension of the last linear layer in the discriminator unchanged, we insert a 2×2 adaptive average pooling layer [13] before the last linear layer.

3.6. SinGAN with Positional Encoding

As discussed in Sec. 3.2, spatial information leaked by zero padding enables SinGAN to capture the global structure and organize various texture patches. However, the implicit positional encoding defined by zero padding introduces unbalanced spatial inductive bias over the whole image space, which always causes unstructured results in the central region of images (Fig. 1(a)). This makes it less ideal for applications that require multi-scale internal sampling, *e.g.*, stretching objects with the main structures retained or with reasonable texture patch recurrence [25, 33].

The spatial anchors in the Cartesian grid can easily keep the global structures fixed in multi-scale sampling, despite that the contents will be interpolated similarly with the resizing mode of the CSG. On the other hand, the scale-invariant transformation between locations in SPE guarantees the organization of patches, which leads to a realistic patch recurrence as in Fig. 1(f). To fulfill different requirements, we adopt the positional encoding in SinGAN by sampling on a positional aligned noise distribution. As the default noise distribution is $\mathcal{N}(0, 1)$, this can be implemented by adding the positional encoding with a sampled noise map.

4. Experiments

Remove Padding. For convolutional generators, the general idea for discarding paddings is to adopt an upsampling layer that interpolates a feature map to a larger size covering the extra padding size for the consecutive convolutional layers. As for the input convolutional block without any upsampling layers, a larger input map will be adopted to avoid additional paddings. As the implicit zero padding in the transposed convolutional layer cannot be removed, following PGGAN [20], we replace it with a bilinear upsampling layer and a convolutional layer.

Architectures and Training Configurations. In addition to internal generative model SinGAN [42], we also study various unconditional generator architectures, including DCGAN [36], PGGAN [20], and StyleGAN2 [22]. All of these methods are trained on 8 Tesla V100 GPUs in PyTorch [35]. We follow their training configurations as closely as possible. In Sec. 4.2, we verify the effectiveness of our MS-PIE in 256^2 StyleGAN2. Three different scales are adopted in MS-PIE with a sampling probability of $[0.5, 0.25, 0.25]$ and the lower resolution is set with the higher probability. Other implementation details are specified in the following sections and the supplementary material.

Table 2: Results based on 256^2 StyleGAN2 with a channel multiplier of two [22] in FFHQ dataset [21, 24]. The Fréchet inception distance (FID) are reported on two best snapshots before the discriminator has been shown with 10M and 20M images, respectively. The precision and recall are reported on the training snapshot with best FID. ↑ indicates higher is better, and ↓ indicates lower is better.

Training configuration	FID@256↓ 10M	FID@256↓ 20M	Precision (%)↑	Recall (%)↑
(a) StyleGAN2-C2-256	6.32	**5.62**	**76.85**	50.41
(b) Deconv → Up-Conv	**5.79**	5.68	76.78	50.46
(c) + Remove padding	6.45	6.13	73.71	51.73
(d) + Cartesian grid	6,31	6.07	73.01	**52.90**
(e) + SPE	6.40	5.86	72.94	**52.87**

Table 3: Multi-level Sliced Wasserstein Distance (SWD) [20] between the synthesized and training images in the 128×128 cropped CelebA dataset. Each column in SWD represents one level of Laplacian pyramid [7] and the last one shows an average of the three distances. ↓ indicates lower is better.

Training configuration	SWD ($\times 10^3$) ↓ 128	64	32	Avg.
(a) PGGAN	**3.162**	**4.285**	**5.000**	**4.149**
(b) + Remove padding	11.169	6.945	7.488	8.534
(c) + SPE, w/o padding	4.555	6.164	6.365	5.694

4.1. Effects of Padding in Existing GANs

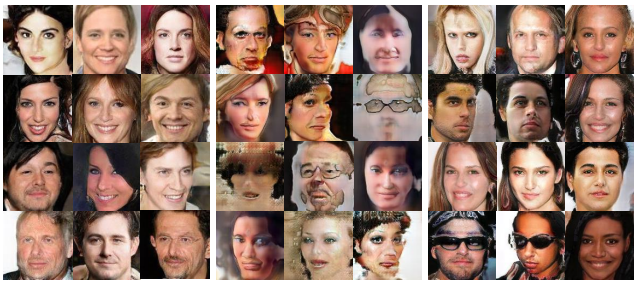
StyleGAN2. In Tab. 2, by measuring Fréchet inception distantance (FID) [15] and the precision and recall [26], we investigate how zero padding influences StyleGAN2 [22] on FFHQ dataset [21]. We first switch to a new baseline Tab. 2(b) where transposed convolutions are replaced with a bilinear upsampling layer and a convolutional layer. This modification yields marginal effects on the final results. Thus, based on it, we conduct the following experiments on padding-free StyleGAN2. In Tab. 2(c), the learnable constant input provides spatial information to the convolutional generator. In Tab. 2(d) and (e), we substitute the constant input with the Cartesian spatial grid and sinusoidal positional encoding, respectively.

As shown in Tab. 2, discarding the implicit positional encoding will directly lead to a higher FID, suggesting that StyleGAN2 actually relies on zero padding to obtain spatial information. On the other hand, the shift from precision to recall indicates that removing padding allows the generator to explore more reasonable spatial structures. Thanks to the spatially consistent transformation between locations, the Cartesian grid and SPE perform better in both FID and recall than the learnable constant input (Tab. 2(c)).

PGGAN and DCGAN. We select two popular generator architectures, *i.e.*, PGGAN [20] and DCGAN [36], to verify that convolutional generators instinctively obtain implicit positional information from zero padding. Table 3 presents the

Table 4: Multi-level SWD [20] between the synthesized and training images in the 128×128 LSUN Bedroom dataset. $\downarrow$ indicates lower is better.

Training configuration	SWD ($\times 10^3$)$\downarrow$			
	128	64	32	Avg.
(a) DCGAN w/ conv	**11.82**	17.30	28.10	19.07
(b) + No padding	22.21	27.26	44.24	31.24
(c) + SPE, w/o padding	14.75	**16.52**	**22.53**	**17.93**



(a) **(b)** **(c)**

Figure 5: Images sampled from various PGGANs trained on cropped CelebA. (a), (b), and (c) indicate the different training configurations in Tab. 3.

results of the more effective PGGAN on highly-structured cropped CelebA [29] dataset. In Tab. 4, to remove zero padding, we use the DCGAN architecture with upsampling and convolutional layers as the baseline in the experiments on LSUN-Bedroom dataset [52].

Unlike the SinGAN and StyleGAN, PGGAN and DCGAN adopt a linear layer on the input noise vector so that the feature map ahead of the convolutional networks contains spatial information. However, as shown in Tab. 3 and Tab. 4, we still observe a significant increase in SWD at each level after removing zero padding in generators. In addition, the padding-free generator fails to synthesize highly structured faces in Fig. 5, which provides convincing evidence that the convolutional generators depend on the implicit positional encoding to obtain spatial information.

To further verify that the lack of positional information causes the higher SWD, we introduce sinusoidal positional encoding (SPE) to the padding-free PGGAN and DCGAN. The SPE is only added with the input feature map ahead of the convolutional generators. As shown in Tab. 3(c) and Tab. 4(c), the explicit positional encoding mitigates the effects of removing implicit spatial inductive bias in each convolutional layer. Thanks to the explicit positional encoding, the synthesized images in Fig. 5(c) can recover the faithful spatial structure. More results and implementation details are presented in our supplementary material.

4.2. MS-PIE in StyleGAN2

In Tab. 5, we examine our MS-PIE in 256^2 StyleGAN2-C2 with multiple image scales of 256^2, 384^2 and 512^2. Results of 384^2 scale point to similar conclusions as with the 512^2 scale, which can be found in our appendix. For the

Figure 6: Multi-scale results from configuration (f) in Tab. 5. The results are in 512^2, 384^2, and 256^2 resolution. The higher resolution is presented with a larger size. Please see our appendix and supplementary video for more results.

512^2 StyleGAN2 baseline in Tab. 5(b), we also downsample the generated images to obtain the FID and P&R at 256^2 scale as another strong baseline.

With zero padding holding spatial anchors, the SPE with 'Expand' resizing mode offers a stable explicit transformation between locations which is unchanged during the multi-scale synthesis. As shown in Tab. 5(f), even if containing fewer upsampling blocks and seeing fewer images at 512^2 scale, the 256^2 StyleGAN2 can still achieve impressive improvement in both FID and P&R. However, once the transformation is changed with the multiple input scales like Tab. 5(d) and Tab. 5(e), there will be a decline in the generation quality at each scale. This phenomenon indicates that the scale-variant transformation between positional encodings prevents the model from easily sharing the learned information among different scales.

Without Padding. Due to the dynamic training scales in our MS-PIE, the spatial anchors are essential for providing unique reference points to directly align the global structure across different image spaces. As shown in Tab 5(h)-(k), abandoning the implicit spatial anchors in each convolutional layer causes an inferior generation quality. However, to some extent, spatial anchors in the Cartesian spatial grid guide the padding-free generator to retain a faithful global structure at multiple scales. Therefore, containing explicit spatial anchors, Tab. 5(i) achieves the least decline in performance among the other positional encodings.

Lite Model. As shown in Tab. 5(g), we select the best configuration (f) and reduce its channel multiplier to investigate the influence of architecture design. Even if we reduce half of the channels in some layers, our MS-PIE enables the lite generator to yield comparable generation quality to the heavy baseline model.

Qualitative Results. Figure 6 presents the multi-scale im-

Table 5: Main results for our MS-PIE with 256^2 StyleGAN2 in the FFHQ dataset. The precision and recall are calculated at the same scale as the Fréchet inception distance (FID).'C2' indicates the channel multiplier in the generator is two.

Training configuration		Resize	FID@512↓ 20M	FID@512↓ 25M	Precision (%)↑	Recall (%)↑	FID@256↓ 20M	FID@256↓ 25M	Precision (%)↑	Recall (%)↑
(a) StyleGAN2-C2-256		—	—	—	—	—-	5.62	5.56	75.92	51.24
(b) StyleGAN2-C2-512		—	3.47	3.41	75.88	54.61	5.00	4.91	75.65	54.58
MS-PIE w/ padding	(c) Leanable constant input	Interp	3.46	3.35	73.84	55.77	4.82	4.50	72.75	55.42
	(d) Cartesian spatial grid	Interp	3.59	3.50	73.28	56.16	**4.74**	4.71	73.34	55.07
	(e) SPE-interp	Interp	3.41	3.15	**74.13**	56.88	4.99	4.73	73.28	**56.93**
	(f) SPE-expand	Expand	**3.31**	**2.93**	73.51	**57.32**	4.79	**4.27**	**73.48**	55.69
	(g) SPE-expand-C1	Expand	3.65	3.40	73.05	56.45	5.54	4.83	73.59	54.29
MS-PIE w/o padding	(h) Learnable constant input	Interp	4.99	4.01	72.81	54.35	5.67	5.11	72.37	55.52
	(i) Cartesian spatial grid	Interp	**3.96**	**3.76**	**73.26**	**54.71**	**5.30**	**5.09**	70.74	56.06
	(j) SPE-interp	Interp	4.80	4.23	73.11	54.63	5.89	5.38	71.21	**56.21**
	(k) SPE-expand	Expand	4.46	4.17	73.05	51.07	6.08	5.59	**72.65**	49.74

Original Image **(a) SinGAN** **(b) SinGAN w/ Cartesian grid** **(c) SinGAN w/ SPE**

Figure 7: Results of SinGAN with different positional embedding strategies. The original image (370×500) is taken from the movie '*Bohemian Rhapsody*' and the sampled images are 1.5 wider than the original image.

ages generated from the best training configuration (f) in Tab. 5. The diverse and realistic multi-scale synthesis further demonstrates the effectiveness of our MS-PIE and the impact of introducing appropriate positional encoding. Importantly, with the help of MS-PIE, the StyleGAN2 that is originally designed for 256^2 generation can also synthesize images in more challenging resolutions, like 896^2 and 1024^2. More training configurations and detailed results in higher resolutions can be found in the supplementary material.

As for image manipulation [1], we customize a convenient pipeline by improving the closed-form factorization [43]. This indicates that our MS-PIE constructs a shortcut for high-resolution image manipulation with a single backbone. The implementation details and high-quality manipulation results are shown in the supplementary material.[4]

4.3. SinGAN with Positional Encoding

This section demonstrates the effectiveness of explicit positional encoding in SinGAN with a challenging case. In Fig. 7, we take a picture from the famous movie '*Bohemian Rhapsody*', where Mercury is singing to thousands of audiences. We use SinGAN to extrapolate the original scene so that Mercy can meet more audiences in a more spacious gym. However, with zero padding offering unbalanced spatial inductive bias, SinGAN can only capture the detailed structure at the borders of the image but generate highly unstructured contents at the center of Fig 7(a).

Different explicit spatial inductive bias enables SinGAN to fulfill various requirements. Adopting the position-aligned noise map with the Cartesian grid, SinGAN can easily interpolate the coarse structure to a larger size, as shown in Fig 7(b). It is the spatial anchors that keep the positions of major contents (Mercury and the tent) unchanged. Meanwhile, due to the scale-variant transformation in Eq. (7), the detailed spatial structure will be stretched, *e.g.*, Mercury accidentally gains a lot of weight. On the contrary, the scale-invariant spatial inductive bias in SPE leads to a more reasonable patch recurrence in Fig. 7(c), while it cannot avoid the shift in the position of Mercury. More results about the versatile image manipulation with positional encoding are presented in the supplementary material.

5. Conclusion

In this work, we thoroughly study how zero padding accidentally encodes imperfect spatial bias for convolutional generators. We have also discussed the strengths of introducing explicit positional encodings, including CSG and SPE, in various existing generator architectures. With the flexible explicit positional encoding, we propose a new multi-scale training strategy (MS-PIE) to achieve high-quality image synthesis at multiple scales with a single 256^2 StyleGAN2. We further show that adopting explicit positional encoding can improve the versatility and robustness of SinGAN.

Acknowledgment. This research was conducted in collaboration with SenseTime. This work is supported by A*STAR through the Industry Alignment Fund - Industry Collaboration Projects Grant. This work is also supported in part by the Early Career Scheme (ECS) through the Research Grants Council (RGC) of Hong Kong under Grant No.24206219, CUHK FoE RSFS Grant, SenseTime Collaborative Grant

[4] YouTube Video, BiliBili Video

References

[1] Rameen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *arXiv preprint arXiv:2008.02401*, 2020. 8

[2] Bilal Alsallakh, Narine Kokhlikyan, Vivek Miglani, Jun Yuan, and Orion Reblitz-Richardson. Mind the pad–cnns can develop blind spots. *arXiv preprint arXiv:2010.02178*, 2020. 2

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *International Conference on Machine Learning*, 2017. 4

[4] Wieland Brendel and Matthias Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. *International Conference on Learning Representations*, 2019. 2

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2018. 4

[6] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 2

[7] Peter Burt and Edward Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983. 6

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 2

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *arXiv preprint arXiv:1810.04805*, 2018. 2, 5

[10] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision*, 2015. 2

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014. 4

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2017. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916, 2015. 6

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017. 6

[16] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[17] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[18] Md Amirul Islam, Sen Jia, and Neil DB Bruce. How much position information do convolutional neural networks encode? *International Conference on Machine Learning*, 2020. 2

[19] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, 2015. 2, 4

[20] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. 1, 4, 6, 7

[21] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 4, 6

[22] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 4, 6

[23] Osman Semih Kayhan and Jan C van Gemert. On translation invariance in cnns: Convolutional layers can exploit absolute spatial location. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[24] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *IEEE International Conference on Computer Vision*, 2014. 6

[25] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: image and video synthesis using graph cuts. *ACM Transactions on Graphics*, 2003. 6

[26] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, 2019. 4, 6

[27] Junghyup Lee, Dohyung Kim, Jean Ponce, and Bumsub Ham. Sfnet: Learning object-aware semantic correspondence. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[28] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, 2016. 2

[29] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *IEEE International Conference on Computer Vision*, 2015. 7

[30] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2016. 3

[31] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *International Conference on Machine Learning*, 2013. 3

[32] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. *arXiv preprint arXiv:2008.02268*, 2020. 2, 5

[33] Tomer Michaeli and Michal Irani. Blind deblurring using internal patch recurrence. In *European Conference on Computer Vision*, 2014. 6

[34] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, 2020. 2, 5

[35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*. 2019. 6

[36] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. 2015. 1, 4, 6

[37] Nasim Rahaman, Aristide 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*, 2019. 2

[38] Adria Recasens, Petr Kellnhofer, Simon Stent, Wojciech Matusik, and Antonio Torralba. Learning to zoom: a saliency-based sampling layer for neural networks. In *European Conference on Computer Vision*, 2018. 2

[39] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015. 2

[40] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6148–6157, 2017. 2

[41] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. End-to-end weakly-supervised semantic alignment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[42] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *IEEE International Conference on Computer Vision*, 2019. 1, 6

[43] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. *arXiv preprint arXiv:2007.06600*, 2020. 8

[44] Assaf Shocher, Shai Bagon, Phillip Isola, and Michal Irani. Ingan: Capturing and remapping the "dna" of a natural image. In *IEEE International Conference on Computer Vision*, 2019. 2

[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 2

[46] 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. *arXiv preprint arXiv:2006.10739*, 2020. 2

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. 2, 4, 5

[48] Jiaqi Wang, Kai Chen, Rui Xu, Ziwei Liu, Chen Change Loy, and Dahua Lin. Carafe: Content-aware reassembly of features. In *IEEE International Conference on Computer Vision*, 2019. 2

[49] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. *International Conference on Learning Representations*, 2018. 2

[50] Rui Xu, Minghao Guo, Jiaqi Wang, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy. Texture memory-augmented deep patch-based image inpainting. *arXiv preprint arXiv:2009.13240*, 2020. 2

[51] Rui Xu, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy. Deep flow-guided video inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[52] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 7

[53] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *European Conference on Computer Vision*, 2016. 2