



Large-size multivariate scale-free texture generation from small-size learning sets with SELF-ATTENTION WASSERSTEIN-GAN

Jhonatan Ancco Cerda, Kinan Abbas, Stéphane G. Roux, Patrice Abry

► To cite this version:

Jhonatan Ancco Cerda, Kinan Abbas, Stéphane G. Roux, Patrice Abry. Large-size multivariate scale-free texture generation from small-size learning sets with SELF-ATTENTION WASSERSTEIN-GAN. 2026. <hal-05801020>

HAL Id: hal-05801020

<https://hal.science/view/index/docid/5801020>

Preprint submitted on 15 Oct 2026

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Copyright - All rights reserved

LARGE-SIZE MULTIVARIATE SCALE-FREE TEXTURE GENERATION FROM SMALL-SIZE LEARNING SETS WITH SELF-ATTENTION WASSERSTEIN-GAN

Jhonatan Ancco Cerda, Kinan Abbas, Stephane G. Roux and Patrice Abry

CNRS, ENS de Lyon, LPENSL, UMR5672, 69342 Lyon Cedex 07, France
{firstname.lastname}@ens-lyon.fr

ABSTRACT

Multivariate scale-free texture images are nowadays central in numerous modern applications very different in nature. A common challenging issue consists of being able to generate such multivariate textures at any arbitrary size, from a learning set of (possibly much) smaller size. Parametric models often turn out to be too restrictive for the generation of realistic textures, thus prompting neural network training-based generation. While several neural network approaches propose generation at sizes larger than training, they do not explicitly assess the quality of the generation by quantifying how well multiscale or scale-free statistics in textures are reproduced and rather rely on perceptual assessment. The present work combines convolutional Wasserstein-GAN architectures with self-attention layers to generate textures of arbitrary size while focusing on reproducing multiscale statistics, without coding a model into the loss function. The relevance of using a self-attention layer is studied. Quantitative assessment conducted on multifractal textures, representative of a broad class of multiscale textures, over ten independent trainings shows that the proposed architecture is able to generate textures with accurate statistics for sizes 8×8 larger than the originals. The results show that placing self-attention in the critic provides the most consistent performance across output sizes.

Index Terms— Wasserstein GAN, self-attention layer, large size extrapolation, scale-free textures

1. INTRODUCTION

Context. Textures, i.e., images with no geometry and thus characterized by their joint statistical distributions, have always been considered essential in image processing. Nowadays, in most recent and modern applications, with the multiplication of sensors, most textures are multivariate (several measured textures characterize the system of interest) and multiscale (their statistics follow some evolution rule with respect to the analysis scale). For instance, multivariate scale-free textures have statistics that are jointly covariant under dilation of the analysis scale [2]. Classical stochastic process-based parametric models usually suffer from not being versatile enough for the generation of rich enough textures, often limited to Gaussian statistics, or too rigid non Gaussian models. This triggered massive neural network training based generation, yet not often considering situations when only small size ground truth images are available. Further, most works do not usually explicitly seek to reproduce multiscale or scale-free non Gaussian statistics and rely

mostly on perceptual rather than quantitative performance assessment. The present work aims to address such limitations.

Related work. Generative Adversarial Networks (GAN), introduced in [3], marked a major advance in image synthesis and demonstrated strong performance in capturing geometric and semantic structures [4]. GAN for texture generation has also been explored [5]. The synthesis of spatially extended textures from observations available only on a smaller spatial support has a long history. Classical non-parametric methods construct arbitrarily large realizations by resampling pixels or patches from a reference texture [6]. Neural approaches replaced explicit resampling with learned representations [7], while convolutional GAN permit generation with arbitrary spatial dimensions [8, 9, 10]. Most of these works, however, rely on local or fixed-scale statistics [11, 12] and are thus not well suited to reproducing multiscale statistics.

Recently, diffusion models have renewed interest in generation beyond the spatial dimensions of the training set [13, 14, 15]. However, much of such works focus on geometric images, do not consider multiscale textures (see a contrario [16]) and rely on per component (rather than multivariate) perceptual quality, semantic consistency, and avoidance of duplicated objects.

Goals, Contributions, and Outline. The goal of this work is to design and assess fully convolutional Wasserstein-GAN architectures trained on images of size $N_T \times N_T$ and able to generate multivariate scale-free textures of arbitrary size $N \times N$, with $N > N_T$. The main contribution is to assess the ability of these architectures to preserve multiscale and cross-component statistical properties when generating textures larger than those used for training, and to study the impact of placing self-attention in the generator, in the critic, or in both networks. The proposed architectures are detailed in Section 2.

In this work, multivariate multifractal textures are used as reference models for scale-free textures because they exhibit strongly non-Gaussian statistics, rich component-wise distributions, and complex inter-component dependencies. They are not considered here for their specific applications, but rather as well-documented reference models for multivariate multiscale textures. Their definitions and analysis are recalled in Section 3. Their statistical properties are assumed not known and not involved in the loss used for training. Section 4 presents the assessment protocol and compares the statistical performance of the different self-attention placements.

2. SELF-ATTENTIVE WGAN

Wasserstein GAN. GAN train a generator G to transform a random input z into synthetic samples, while a discriminator learns to distinguish generated samples from training samples [3]. Wasserstein GAN (WGAN) replace the discriminator with a critic C , known to

The support of the Centre Blaise Pascal IT test platform at ENS de Lyon (France) for Machine Learning facilities is gratefully acknowledged. The platform operates the SIDUS solution developed by E. Quemener [1].

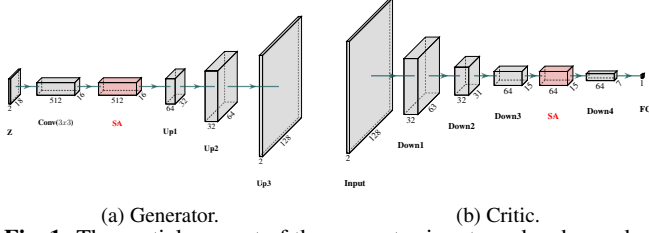


Fig. 1. The spatial support of the generator input can be changed at inference. Red blocks indicate the two candidate self-attention (SA) locations evaluated in this work: after the first generator convolution and between the third and fourth critic convolutions.

provide a more stable training framework [17, 18]. For training the critic, a loss function entailing a gradient penalty enforcing the Lipschitz constraint is used,

$$\mathcal{L}_C = -\mathbb{E}_{x \sim p_{\text{data}}} [C(x)] + \mathbb{E}_{z \sim p_z} [C(G(z))] + \lambda_{\text{GP}} \mathbb{E}_{\tilde{y} \sim p_{\tilde{y}}} [\max(0, \|\nabla_{\tilde{y}} C(\tilde{y})\|_2 - 1)^2], \quad (1)$$

where $\tilde{y}$ denotes an interpolation between real and generated samples, and λ_{GP} controls the gradient penalty. The generator is trained to maximize the critic score assigned to generated samples, or equivalently to minimize

$$\mathcal{L}_G = -\mathbb{E}_{z \sim p_z} [C(G(z))]. \quad (2)$$

Spatial-support extrapolation. The classical WGAN architecture suffers from a major restriction: the fully connected layer, used as the first layer of the generator, imposes that generated textures have one same and unique size, that of the training set.

For univariate unidimensional time series, it was proposed in [10] to replace that fully-connected input layer by a convolutional layer. This results in a convolutional generator for which the spatial support of the input noise can be changed at inference time: The learned parameters can be used to generate samples on supports larger than those available during training.

We extend this principle here from one-dimensional univariate time series to two-dimensional multivariate textures. The present work explores the potential critical impact of the convolutional kernel size in the performance of the proposed generator. Compared performance are reported and discussed in Section 4.

Self-attention layer. In addition, we devise richer generator/critic architectures by incorporating self-attention layers. While convolutional layers combine information within local neighborhoods, self-attention layers combine learned features from different spatial locations, enabling nonlocal interactions. Given a feature map $\mathbf{F}$, three learned 1×1 convolutions produce query, key, and value matrices $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$. The self-attention output is

$$\tilde{\mathbf{F}} = \mathbf{F} + \alpha \text{softmax} \left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} \right) \mathbf{V}, \quad (3)$$

where d is the query/key dimension and α is a learned scalar.

In the generator, self-attention is inserted immediately after the first convolutional layer, where the spatial support is smallest during training. In the critic, it is inserted between the third and fourth convolutional layers, where the features combine information from broader receptive fields. These positions limit the quadratic cost of attention while allowing nonlocal information to be processed before the subsequent generator or critic layers [19, 20].

Proposed architecture. The common backbone combines a convolutional generator and a convolutional critic, as illustrated in Fig. 1.

To balance representation capacity and computational cost, the channel depth progressively decreases in the generator and increases in the critic. The generator uses convolutional upsampling blocks with batch normalization and ReLU activations, whereas the critic uses stride-(2, 2) convolutions with leaky-ReLU activations for spatial downsampling. The generator and critic backbones contain $\simeq 323\text{k}$ and $\simeq 68\text{k}$ parameters, respectively, excluding the optional self-attention blocks. For comparison, a generator with a fully connected first layer would require approximately 3576k parameters to generate images of size $64 \times 64 \times 3$ [21].

For clarity, Fig. 1 displays the self-attention blocks in both the generator and the critic. Their separate and joint placements are assessed in Section 4.

3. MULTIVARIATE MULTISCALE TEXTURES

Multivariate scale-free texture models. M-variate multifractal random walks (MMRW), used as rich multiscale multivariate stochastic models for textures, are defined as [2, 22]:

$$\mathbf{X}(\mathbf{x}) = (X^{(1)}(\mathbf{x}), \dots, X^{(M)}(\mathbf{x})) \text{ where}$$

$$X^{(m)}(\mathbf{x}) = e^{\omega_{\lambda_m}^{(m)}(\mathbf{x})} G_{H_m}^{(m)}(\mathbf{x}), \quad (4)$$

and with $\{G_{H_m}^{(m)}(\mathbf{x})\}_{m=1}^M$ and $\{\omega_{\lambda_m}^{(m)}(\mathbf{x})\}_{m=1}^M$ are jointly Gaussian zero-mean fields.

Technically, $G_{H_m}^{(m)}(\mathbf{x}) m = 1^M$ denote multivariate fractional Gaussian noise (fGn), characterized by the Hurst parameters $H_m m = 1^M$, which control the cross-spatial correlations [5, 23]:

$$\mathbb{E} \{ G_H^{(m)}(\mathbf{x}) G_H^{(m')}(\mathbf{x}') \} = \rho_{m,m'} C_H(\mathbf{x} - \mathbf{x}'), \quad (5)$$

where $C_H(\cdot)$ denotes the covariance function of the univariate fGn with Hurst exponent $H = (H_m + H_{m'})/2$, and $\rho_{m,m'} \in [-1, 1]$ defines the cross-correlation coefficients.

Expanding on univariate definition [22], the $\{\omega_{\lambda_m}^{(m)}(\mathbf{r})\}_{m=1}^M$ are further defined via their auto- and cross-correlation as:

$$\mathbb{E} \{ \omega_{\lambda_m}^{(m)}(\mathbf{x}) \omega_{\lambda_{m'}}^{(m')}(\mathbf{x}') \} = \rho_{m,m'}^{mf} \lambda_m \lambda_{m'} \log \left(\frac{L}{\|\mathbf{x} - \mathbf{x}'\| + 1} \right), \quad (6)$$

for $\|\mathbf{x} - \mathbf{x}'\| \leq L$, and zero otherwise, $\rho_{m,m'}^{mf} \in [-1, 1]$ defines cross-correlation coefficients [24, 5].

Intuitively, the $\{G_{H_m}^{(m)}(\mathbf{x})\}_{m=1}^M$ constitute jointly Gaussian textures with algebraically decaying (cross-) correlations thus inducing long-range correlations, while the $\{\omega_{\lambda_m}^{(m)}(\mathbf{x})\}_{m=1}^M$ act as covariance spatially structured modulations, inducing strong departures from joint Gaussianity in MMRW, explicitly controlled jointly by λ_m and $\rho_{m,m'}^{mf}$ while preserving scale-free spatial structures at all statistical orders. This is why $\rho_{m,m'}^{mf}$ is referred to as the multifractal correlation, with a slight abuse of language as it quantifies higher order statistics [24, 5].

Multiscale (wavelet-based) analysis. It is well-documented that scale-free and multifractal textures are well-characterized using wavelet-based multiscale representations [25]. Let $\{d_{X^{(m)}}(j, \mathbf{k})\}_{m=1}^M$ denote the M-variate coefficients of a discrete wavelet transform computed from multivariate texture X , defined as inner products between each $X^{(m)}$ and dilated and translated wavelet templates at scale 2^j and spatial location $2^j \mathbf{k}$ [26]. To robustly capture scale-free behaviors at all statistical orders, hence, multifractal properties,

wavelet leaders are defined as local suprema of wavelet coefficients taken over finer scales and within a spatial neighborhood: $L_{X^{(m)}}(j, \mathbf{k}) = \sup_{j' \leq j, \mathbf{k}' \in [k2^j, (k+1)2^j]} d_{X^{(m)}}(j', \mathbf{k}') [25, 5]$.

For MMRW, the first order cumulants of the logarithm of the leaders characterizes the univariate scale-free statistics of each component via linear behaviors as function of scales:

$$C_m^{(1)}(j) = \mathbb{E} \log L_{X^{(m)}}(j, \mathbf{k}) = c_m^{(1,0)} + c_m^1 \log 2^j, \quad (7)$$

$\forall m = 1, \dots, M$, with $c_m^1 = H_m + \lambda_m^2/2$, thus mostly dominated by the correlation structure and primarily reflecting the (univariate) second-order statistics [22, 5]. The higher order cumulants of the logarithm of the leaders reflects the cross-higher-order statistics of the process. Notably, the second order cumulants characterizes the scale-free cross-statistics via a linear behaviors as function of scales:

$$\begin{aligned} C_{m,m'}^{(2)}(j) &= \text{Var} \log L_{X^{(m)}}(j, \mathbf{k}) \log L_{X^{(m')}}(j, \mathbf{k}) \\ &= c_{m,m'}^{(2,0)} + c_{m,m'}^2 \log 2^j, \end{aligned} \quad (8)$$

$\forall m, m' = 1, \dots, M$, with $c_{m,m'}^2 = -\rho_{m,m'}^{mf} \lambda_m \lambda_{m'}$ quantifying cross multifractality and intermittency [22, 5].

To further quantify the cross-statistics, one usually defines the wavelet covariance function as [27, 24, 5], with $S_{m,m'}(j) = \sum_k d_{X_m}(j, k) d_{X_{m'}}(j, k)$:

$$\rho_{m,m'}^{wav}(j) = \frac{S_{m,m'}(j)}{\sqrt{S_{m,m}(j) S_{m',m'}(j)}}. \quad (10)$$

For MMRW, $\rho_{m,m'}^{wav}(j)$ should not depend on j and be equal to $\rho_{m,m'}$: $\rho_{m,m'}^{wav}(j) = \rho_{m,m'}, \forall j > 0$, a property that can be used to assess the quality of the generated textures.

In addition, one can compute a multifractal counterpart to the wavelet covariance function as:

$$\rho_{m,m'}^{mf}(j) = \frac{C_{m,m'}^{(2)}(j)}{\sqrt{C_{m,m}^{(2)}(j) C_{m',m'}^{(2)}(j)}}. \quad (11)$$

For MMRW, $\rho_{m,m'}^{mf}(j)$ should converge to $\rho_{m,m'}^{mf}$ in the limit of fine scales $2^j \rightarrow 0$. While this asymptotic limit is not reachable in practice, the function $\rho_{m,m'}^{mf}(j)$ can still be used for comparisons between textures from the generated and training datasets, as a means to assess that cross dependencies beyond second-order statistics (cross-correlations) are satisfactorily reproduced and for estimating $\rho_{m,m'}^{mf}$ by averaging $\rho_{m,m'}^{mf}(j)$ across fine scales.

4. PERFORMANCE ASSESSMENT

4.1. Experimental setup

Training dataset. The dataset consists of 10,000 independent bivariate ($M = 2$) MMRW images of size 128×128 , of which 8,000 are used for training and 2,000 for validation. The images are generated using the circular matrix embedding toolbox described in [28]. The two components have different marginal multifractal properties, with self-similarity parameters $H = [0.8, 0.6]$ and multifractality parameters $\lambda = [\sqrt{0.02}, \sqrt{0.04}]$. Inter-component dependencies are set to $\rho_{1,2} = -0.4$ for second-order correlations and $\rho_{1,2}^{mf} = 0.6$ for cross-multifractal dependencies. This setting is chosen because generating MMRW with different multifractal structures for each component is more demanding than the settings considered in [5, 10], since the neural network must jointly reproduce distinct component-wise multiscale properties and inter-component dependencies. Also,

the dependence structure is challenging as the second order correlation is negative (global anti-correlation) while the multifractal correlation is positive (positively correlated transient or local structures).

Training settings. Four configurations are considered: a WGAN without self-attention (W), self-attention in the generator only (SA-G), self-attention in the critic only (SA-C), and self-attention in both networks (SA-GC). All configurations use the same training settings. We use Adam with learning rate 10^{-4} , $\beta_1 = 0$, and $\beta_2 = 0.9$, a gradient penalty coefficient $\lambda_{GP} = 10$, and a batch size of 8. Five critic updates are performed for each generator update. All configurations are trained for 25 epochs.

Evaluation protocol. The networks are trained using 128×128 textures only. At inference, the spatial size $N_z \times N_z$ of the input noise is varied to generate textures of size $N \times N$, with $N = 128, 256, 512, 1024$, and 2048. The corresponding input noise sizes are $2 \times N_z \times N_z$, with $N_z = 18, 34, 66, 130$, and 258, respectively. For each configuration, training run, and output size, 25 textures are generated. The multiscale statistics introduced in Section 3 are computed and used to estimate c_1, c_2, ρ , and ρ^{mf} . To account for training variability, the complete experiment is repeated over 10 independent trainings from scratch, using the same training set and training conditions. Bias, RMSE, and standard deviation are computed for each training and then averaged over the 10 trainings.

4.2. Results and Discussion

Overfitting assessment. To check that the generator does not simply reproduce training samples, each generated texture is compared with translated and rotated versions of the training textures, and the largest absolute Pearson correlation is retained. The same procedure is applied to the validation set. Fig. 2 shows nearly identical distributions, indicating no memorization.

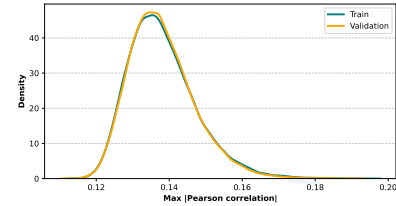


Fig. 2. Distributions of the maximum absolute Pearson correlations with translated and rotated training textures, computed for SA-C-generated textures and validation textures.

Architecture design and complexity. A comparison of $3 \times 3, 9 \times 9$, and 15×15 first-layer convolutional kernel sizes showed that the 3×3 kernel provided the best overall validation performance while requiring the lowest parameter and computational costs. Detailed results are omitted due to space constraints, and all reported performances use the 3×3 kernel. This indicates that a small receptive field is sufficient to reproduce local bursty bivariate structures.

Spatial-support extrapolation performance. Table 1 compares the statistical performance of the four attention configurations over the different generated texture sizes. Among the 30 combinations of output size and statistical descriptor, SA-C achieves the lowest RMSE in 15 cases and ties for the lowest RMSE in one additional case. Its advantage is particularly clear at $N = 1024$, where it provides the lowest RMSE for five of the six descriptors. The models generate textures up to $N = 1024$, eight times the training size along each spatial dimension, with good overall agreement across the evaluated descriptors. At $N = 2048$, the RMSE increases for the first-order scaling coefficients and the wavelet correlation in all configurations.

Table 1. Generation performance for the four model families (W, SA-C, SA-G, SA-CG), in terms of RMSE, Bias and standard deviations averaged across 10 independent trainings. In addition, $V(B)$ and $V(R)$ report the variability across independent trials in terms of std of the Biases and RMSE. **Red** indicates the overall best among all four models; **green** indicates the best among SA-C, SA-G, SA-CG when W is better; gray indicates the other two SA variants.

N	Metric	$c_1^{(1)} [10^{-2}]$				$c_1^{(2)} [10^{-2}]$				$c_2^{(1)} [10^{-2}]$				$c_2^{(2)} [10^{-2}]$				$\rho^{wav} [10^{-2}]$				$\rho^{mf} [10^{-2}]$			
		W	SA-C	SA-G	SA-CG	W	SA-C	SA-G	SA-CG	W	SA-C	SA-G	SA-CG	W	SA-C	SA-G	SA-CG	W	SA-C	SA-G	SA-CG	W	SA-C	SA-G	SA-CG
128	RMSE	10.40	8.43	9.65	9.06	8.66	8.51	11.33	9.80	3.54	2.94	3.66	2.94	4.07	3.52	4.72	4.92	3.83	4.42	5.08	5.77	30.8	28.5	32.3	28.2
	Bias	-0.18	-1.76	1.74	-1.42	2.74	2.46	5.69	4.38	0.47	-0.38	0.76	-0.21	0.74	0.68	0.73	2.74	1.26	2.15	2.46	3.68	-28.9	-26.8	-30.4	-26.6
	Std	8.67	8.13	9.02	8.26	7.58	8.05	8.89	8.50	3.52	2.83	3.46	2.94	3.98	3.40	4.28	4.09	3.11	3.64	3.93	4.50	10.8	9.5	10.9	9.6
	$V(B)$	6.80	2.17	3.60	4.26	4.07	2.39	6.44	3.29	0.76	1.02	1.19	0.58	1.19	1.02	2.60	1.15	2.21	1.75	2.33	1.13	4.2	2.9	4.5	3.4
	$V(R)$	2.89	1.05	1.57	2.43	2.22	1.67	4.90	2.01	0.77	0.59	0.51	0.58	1.18	0.59	2.31	1.37	1.04	1.10	0.74	1.35	3.8	2.8	4.2	3.5
256	RMSE	5.82	4.17	7.11	5.16	7.59	6.92	7.45	6.88	2.19	1.36	1.82	1.30	2.01	2.16	2.25	2.87	3.33	3.40	5.13	6.06	32.1	29.9	34.1	31.6
	Bias	1.83	-0.67	1.67	-1.61	6.03	5.26	4.76	4.62	0.98	0.20	0.22	0.19	0.89	1.16	0.57	2.08	0.97	1.82	2.99	4.84	-31.4	-29.5	-33.2	-30.9
	Std	4.49	3.83	5.62	4.33	3.97	4.28	5.20	4.77	1.90	1.24	1.48	1.24	1.68	1.71	1.72	1.93	2.55	2.65	3.68	3.60	6.7	5.1	7.7	6.3
	$V(B)$	4.06	1.94	5.71	2.74	3.79	2.37	3.62	2.80	0.88	0.63	1.34	0.44	0.85	0.91	1.66	0.90	2.39	1.64	2.81	1.83	4.4	2.0	5.7	4.0
	$V(R)$	2.40	0.95	4.49	1.17	2.84	1.74	2.73	2.05	1.29	0.27	0.85	0.18	0.50	0.62	1.01	0.81	1.21	1.12	2.23	1.86	4.4	2.0	5.8	3.8
512	RMSE	3.59	3.98	7.52	4.08	4.32	5.15	5.86	4.77	1.19	0.85	1.21	0.90	1.15	1.64	1.46	2.06	3.03	2.76	5.18	8.76	32.9	31.9	35.2	32.9
	Bias	0.32	3.02	0.94	-1.74	2.79	4.27	2.65	2.04	0.73	0.55	0.45	0.47	0.78	1.33	0.49	1.62	0.27	0.78	2.41	7.99	-32.6	-31.7	-34.8	-32.6
	Std	2.68	2.33	4.40	2.60	2.13	2.62	3.91	3.01	0.88	0.59	0.85	0.69	0.76	0.91	0.97	0.96	1.98	1.90	3.00	3.62	4.1	3.7	5.1	4.4
	$V(B)$	2.87	1.88	7.28	3.33	4.43	2.26	4.33	3.65	0.58	0.63	0.92	0.43	0.69	0.62	1.37	1.11	2.60	2.33	5.43	4.27	4.4	1.8	5.6	6.1
	$V(R)$	1.33	1.43	3.84	1.83	3.53	1.87	2.52	1.75	0.45	0.23	0.52	0.23	0.56	0.55	0.90	0.71	1.05	1.28	4.26	4.60	4.3	1.7	5.5	5.9
1024	RMSE	3.44	2.20	8.26	7.08	4.41	2.60	6.05	4.40	0.71	0.46	0.66	0.53	0.96	1.42	1.11	2.04	3.88	3.69	6.82	12.96	34.0	32.5	35.5	33.6
	Bias	-2.94	-1.54	-0.85	-6.84	-2.47	-0.82	0.69	-2.19	0.54	0.36	0.22	0.24	0.87	1.34	0.61	1.41	-0.71	1.53	1.11	12.30	-33.9	-32.4	-35.2	-33.5
	Std	1.50	1.10	3.91	1.71	1.35	1.53	3.30	2.43	0.43	0.24	0.46	0.33	0.36	0.45	0.58	0.67	1.58	1.65	3.16	3.97	2.7	2.4	3.8	3.3
	$V(B)$	1.84	1.78	8.50	3.49	4.20	2.45	5.27	3.86	0.34	0.26	0.48	0.39	0.45	0.47	0.91	1.57	4.05	3.97	8.50	7.51	3.6	1.6	6.1	7.2
	$V(R)$	1.50	1.32	4.46	3.44	2.20	1.36	2.27	2.49	0.30	0.21	0.19	0.18	0.40	0.44	0.51	0.80	1.76	2.50	5.92	7.85	3.6	1.6	6.1	7.1
2048	RMSE	9.55	10.44	9.76	15.13	8.64	6.71	6.34	7.88	0.44	0.34	0.45	0.39	1.16	1.54	1.02	2.11	4.95	5.07	9.22	18.87	34.7	33.4	35.2	34.1
	Bias	-9.50	-10.42	-6.85	-15.08	-8.41	-6.63	-3.67	-7.42	0.35	0.31	0.03	0.18	1.14	1.51	0.73	1.46	-1.33	3.25	0.31	18.39	-34.7	-33.4	-35.0	-34.0
	Std	0.97	0.70	2.64	1.28	0.91	1.01	2.43	1.74	0.22	0.13	0.26	0.19	0.22	0.25	0.36	0.53	1.24	1.48	3.20	3.99	1.7	1.7	3.1	2.7
	$V(B)$	2.19	1.54	8.68	3.60	3.83	2.24	5.75	3.58	0.27	0.18	0.42	0.36	0.32	0.38	0.79	1.65	5.59	5.54	11.34	11.50	3.6	1.3	6.8	7.6
	$V(R)$	2.17	1.53	5.73	3.61	3.34	2.22	3.60	2.90	0.21	0.16	0.18	0.21	0.31	0.37	0.45	0.83	2.76	4.04	7.03	11.63	3.6	1.2	6.7	7.5

Attention in the generator does not provide the same improvement. SA-G presents larger variability across independent trainings and degrades the scaling and cross-correlation statistics as the generated size increases, particularly from $N = 1024$. A possible explanation is that generator attention is trained on 16×16 feature maps but is applied to increasingly larger feature maps during extrapolation. The resulting size-dependent mixing may distort the scaling and cross-component correlations. In contrast, critic attention is used only during training at the fixed training size and does not modify the generated features at inference.

Beyond the aggregate errors, Fig. 3 compares the scale-by-scale statistical behavior of W and SA-C. Both configurations reproduce the approximately linear evolution of $C_1(j)$ and $C_2(j)$ up to $N = 1024$. The deviations observed at $N = 2048$ confirm the extrapolation limit identified in Table 1.

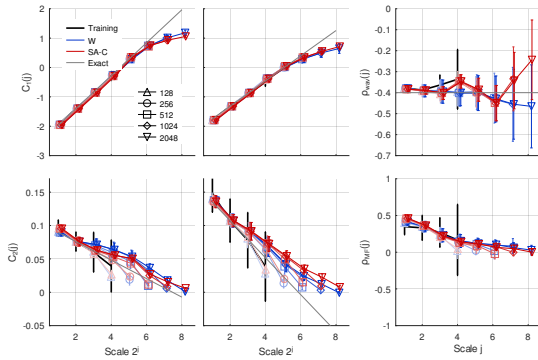


Fig. 3. Scale-wise comparison of W and SA-C for the different generated texture sizes.

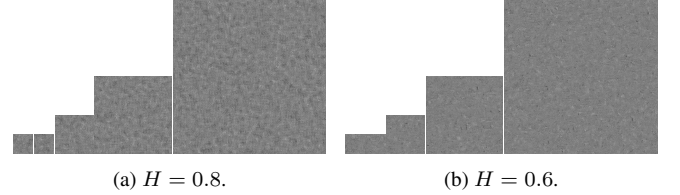


Fig. 4. Training textures at $N_T = 128$ followed by SA-C-generated textures at $N = 128, 256, 512$, and 1024 , from left to right, for the two components with $H = 0.8$ and $H = 0.6$.

Finally, Fig. 4 shows that the SA-C generator trained at $N_T = 128$ produces visually consistent textures at increasing spatial supports.

5. CONCLUSIONS AND PERSPECTIVES

This work addressed the generation of multivariate multifractal scale-free textures on spatial supports larger than those available during training. A fully convolutional WGAN trained on 128×128 textures reproduces the targeted multiscale and cross-statistical properties up to 1024×1024 , without including these properties in the training loss. Among the evaluated self-attention placements, critic-only attention provides the most consistent performance while introducing no additional cost during generation. Performance degradation at 2048×2048 indicates the current extrapolation limit. Future work will consider higher-dimensional textures and statistically informed learning objectives.

6. REFERENCES

- [1] E. Quemener and M. Corvellec, “Sidus—the solution for extreme deduplication of an operating system,” *Linux J.*, vol. 2013, no. 235, Nov. 2013.
- [2] B. B. Mandelbrot, “Intermittent turbulence in self-similar cascades: divergence of high moments and dimension of the carrier,” *J. Fluid Mech.*, vol. 62, pp. 331–358, 1974.
- [3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Adv. Neural Inf. Process. Syst.*, vol. 27, 2014.
- [4] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4401–4410.
- [5] P. Abry, V. Mauduit, E. Quemener, and S. Roux, “Multivariate multifractal texture dcgan synthesis: How well does it work? how does one know?,” *J. Signal Process. Syst.*, pp. 1–17, 2022.
- [6] A. Efros and K. Leung, “Texture synthesis by non-parametric sampling,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1999, pp. 1033–1038.
- [7] L. Gatys, S. Ecker, and Matthias Bethge, “Texture synthesis using convolutional neural networks,” in *Advances in Neural Information Processing Systems*, 2015, vol. 28, pp. 262–270.
- [8] D. Ulyanov, V. Lebedev, Andrea V., and V. Lempitsky, “Texture networks: Feed-forward synthesis of textures and stylized images,” in *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016, vol. 48 of *Proceedings of Machine Learning Research*, pp. 1349–1357.
- [9] U. Bergmann, N. Jetchev, and R. Vollgraf, “Learning texture manifolds with the periodic spatial GAN,” in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017, vol. 70 of *Proceedings of Machine Learning Research*, pp. 469–477.
- [10] T. Beroud, P. Abry, Y. Malevergne, M. Senneret, G. Perrin, and J. Macq, “Wassertein gan synthesis for time series with complex temporal dynamics: frugal architectures and arbitrary sample-size generation,” in *ICASSP 2023. IEEE*, 2023, pp. 1–5.
- [11] A. Shocher, S. Bagon, P. Isola, and M. Irani, “InGAN: Capturing and retargeting the “DNA” of a natural image,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 4492–4501.
- [12] R. Shaham, T. Dekel, and T. Michaeli, “SinGAN: Learning a generative model from a single natural image,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 4570–4580.
- [13] V. Kulikov, S. Yadin, M. Kleiner, and T. Michaeli, “SinDDM: A single image denoising diffusion model,” in *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023, vol. 202 of *Proceedings of Machine Learning Research*, pp. 17920–17930.
- [14] Y. He, S. Yang, H. Chen, X. Cun, M. Xia, Y. Zhang, X. Wang, R. He, Q. Chen, and Y. Shan, “ScaleCrafter: Tuning-free higher-resolution visual generation with diffusion models,” in *International Conference on Learning Representations (ICLR)*, 2024.
- [15] R. Du, D. Chang, T. Hospedales, Y. Song, and Z. Ma, “Demo-Fusion: Democratising high-resolution image generation with no \$\$\$,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 6159–6168.
- [16] K. Abbas, P. Abry, and S. Roux, “Diffusion Models for Multifractal Texture Synthesis,” in *The 33rd European Signal Processing Conference (EUSIPCO 2025)*, Palermo (Italie), Italy, Sept. 2025.
- [17] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *International conference on machine learning*. Pmlr, 2017, pp. 214–223.
- [18] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, “Improved training of wasserstein gans,” *Advances in neural information processing systems*, vol. 30, 2017.
- [19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [20] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, “Self-attention generative adversarial networks,” in *International conference on machine learning*. Pmlr, 2019, pp. 7354–7363.
- [21] Alec Radford, Luke Metz, and Soumith Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” in *International Conference on Learning Representations (ICLR)*, 2016.
- [22] E. Bacry, J. Delour, and J.-F. Muzy, “Multifractal random walk,” *Physical Review E*, vol. 64, no. 2, pp. 026103, 2001.
- [23] Charles-Gérard Lucas, Gustavo Didier, Herwig Wendt, and Patrice Abry, “Multivariate selfsimilarity: Multiscale eigenstructures for selfsimilarity parameter estimation,” *IEEE Transactions on Signal Processing*, vol. 72, pp. 1739–1749, 2024.
- [24] R. Leonarduzzi, P. Abry, S. G. Roux, H. Wendt, S. Jaffard, and S. Seuret, “Multifractal characterization for bivariate data,” in *Proc. European Signal Processing Conference (EUSIPCO)*, Rome, Italy, September 2018.
- [25] H. Wendt, P. Abry, and S. Jaffard, “Bootstrap for empirical multifractal analysis,” *IEEE Signal Processing Magazine*.
- [26] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, San Diego, CA, 1998.
- [27] Brandon Whitcher, Peter Guttorp, and Donald B Percival, “Wavelet analysis of covariance with application to atmospheric time series,” *Journal of Geophysical Research: Atmospheres*, vol. 105, no. D11, pp. 14941–14962, 2000.
- [28] H. Helgason, V. Pipiras, and P. Abry, “Fast and exact synthesis of stationary multivariate gaussian time series using circulant embedding,” *Signal Processing*, vol. 91, no. 5, pp. 1123–1133, 2011.