Diffusion Models for Multifractal Texture Synthesis

Kinan Abbas[1,2], Patrice Abry[1] and Stephane Roux[1]
[1] *CNRS, ENS de Lyon, LPENSL*, UMR5672, 69342, Lyon cedex 07, France
[2] *XMBAUBLE*, Lyon, France
{firstname.lastname}@ens-lyon.fr

***Abstract*—Multifractal textures provide a robust framework for modeling real-world textures characterized by complex, transient, and statistically rich patterns, with applications spanning biomedical imaging to material science. While diffusion models have proven effective in generative tasks, their ability to synthesize textures, i.e., images with no geometry but instead with rich and complex spatial dynamics, remains underexplored. This study investigates the performance of diffusion neural networks in generating multifractal textures, which are used as representatives of such complex textures with prescribed statistical properties, yet without embedding multifractal information in the training loss. From a dataset of 1000 multifractal textures, a U-Net-based diffusion model is trained, under four different noise schedulers to explore their influence on synthesis quality. Performance is evaluated by comparing the multifractal statistics, assessed by wavelet-leader analysis, of generated textures against those of the training set. Results suggest that overall the linear noise scheduler performs best in reproducing multifractal properties in textures.**

***Index Terms*—Diffusion Models, Multifractal Textures, Texture Synthesis, Wavelet-Leader Analysis, Noise Schedulers**

I. Introduction

Context. Neural network (NN)-based image generative models have become critical in a wide variety of applications [1]. However, most images generated by NNs contain high geometric content (e.g., faces, buildings, streets) [2], rather than pure texture images with little or no geometry, characterized instead by rich and complex spatial dynamics and intricate statistical properties. The major goal of this work is to investigate whether Diffusion Networks (DN), the most recent and promising paradigm in generative Artificial Intelligence (AI) [3], can efficiently synthesize multifractal textures, used here as representatives of textures with rich multiscale properties.

Related work. Generative Adversarial Networks (GANs), first proposed by Goodfellow et al. [1], marked a major advancement in image generation. They are capable of capturing geometric structures but often fail to preserve fine statistical details [4]. Diffusion models, pioneered by Sohl-Dickstein et al. [5] and further developed in [3], have since emerged as a superior alternative offering improved control over statistical properties. These models iteratively transform noise into structured outputs by reversing a stochastic diffusion process. Subsequent advancements, such as denoising diffusion probabilistic models [3] and score-based generative methods [6], have refined their efficiency and applicability, particularly in image generation. Unlike GANs, diffusion models better capture fine-grained statistical properties, making them well-suited for various applications [7].

Diffusion models have recently been adapted for texture synthesis. Chen et al. [8] utilized diffusion models for text-driven 2D texture synthesis, generating visually rich patterns from textual descriptions. Similarly, Cao et al. [9] introduced a method for 3D textures, synthesizing surface details for geometric models using image diffusion techniques. Yu et al. [10] further showcased their potential in generating mesh textures, providing a generative solution for 3D graphics applications. Despite these advances, the potential of diffusion models for synthesizing statistically complex textures, such as those with multifractal properties, remains largely unexplored.

In terms of texture models, Gaussian random fields have been widely used in a variety of applications [11]. While they effectively describe the global spatial dynamics of textures by accurately modeling the power spectral density, they are not well suited to capture transient patterns or local structures in textures. Multifractal processes enrich Gaussian random fields as texture models by incorporating rich local or transient patterns, with tight spatial organization, which are not already accounted for in the power spectral density (cf. e.g., [12]).

Goals, Contributions, and Outline. The overarching goal of this work is to quantify the ability of Diffusion Networks to generate multifractal textures, used as representatives of textures with rich and complex spatial dynamics and intricate statistical properties beyond those of Gaussian processes. This evaluation is conducted using a dataset of 1,000 univariate multifractal random walk (MRW) images (256×256 pixels) to assess the effect of four noise schedulers—cosine, linear, quadratic, and sigmoid—on synthesis quality[1]. Results demonstrate that the linear scheduler outperformed the other noise scheduling strategies in consistently preserving the global scaling correlations inherent in the training data while also effectively reproducing the multifractal properties

The paper is organized as follows: Section II introduces theoretical background related to diffusion networks. Section III defines multifractal processes and recalls the key steps of the wavelet-leader-based analysis multifractal texture analysis. Section IV describes the methodology used to assess the ability of Diffusion Networks to generate univariate multifractal textures. Section V details the results and discussion of the synthesis quality and challenges encountered. Finally, Section VI concludes the paper with a summary of the findings and directions for future work.

[1] Source codes will be made available at the time of publication via an open repository.

II. Diffusion Models

Diffusion models represent a class of generative models that synthesize data by iteratively reversing a stochastic noise-adding process, transforming pure noise into structured outputs [5]. The process begins with a forward diffusion stage, defined as a Markov chain over T discrete timesteps. Here, an initial sample $\mathbf{x}_0 \sim q(\mathbf{x}_0)$—e.g., a texture image—is progressively perturbed by Gaussian noise according to

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \tag{1}$$

where $\beta_t \in (0,1)$ is the time-dependent noise variance controlling the noise intensity at each step t. This forward process admits a convenient closed-form expression:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}), \tag{2}$$

with $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t}\alpha_s$, enabling efficient sampling of noisy states directly from the original data.

Diffusion models generate images by reversing the noise corruption process. This reverse step is learned using a neural network, usually a U-Net [3], and is modeled as:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I}), \tag{3}$$

where $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$ predicts the mean of the denoising step, and σ_t^2 is a predefined variance. Training optimizes a simplified variational bound, reducing the mean-squared error loss:

$$\mathbb{E}_{\mathbf{x}_0,\epsilon}[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2], \tag{4}$$

where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the added noise, and $\epsilon_\theta(\mathbf{x}_t, t)$ is the U-Net's noise prediction at timestep t. The choice of noise schedule $\{\beta_t\}_{t=1}^{T}$—whether linear, cosine, quadratic, or sigmoid—further shapes the model's ability to balance detail preservation and statistical accuracy.

The noise schedule controls the rate of noise injection in the forward process and significantly influences the quality and statistical fidelity of the generated output in diffusion models. A *linear schedule* gradually increases β_t from a small value (e.g., 0.0001) to a larger one (e.g., 0.02) over T steps [3]. This simple and consistent noise increase helps maintain a balance between training efficiency and sample diversity:

$$\beta_t = \beta_{\text{start}} + \frac{t-1}{T-1}(\beta_{\text{end}} - \beta_{\text{start}}), \tag{5}$$

where $\beta_{\text{start}} = 0.0001$ and $\beta_{\text{end}} = 0.02$.

In contrast, the *cosine schedule*, proposed by Nichol et al. [13], follows a cosine decay pattern, slowing noise addition near the start and end of the process:

$$\bar{\alpha}_t = \frac{\cos^2\left(\frac{\pi}{2}\cdot\frac{t/T+s}{1+s}\right)}{\cos^2\left(\frac{\pi}{2}\cdot\frac{s}{1+s}\right)}, \quad \beta_t = 1 - \frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}, \tag{6}$$

where $s = 0.008$. *Quadratic schedules* accelerate noise either early or late in the timeline and can be expressed as follows:

$$\beta_t = \left(\beta_{\text{start}}^{1/2} + \frac{t-1}{T-1}(\beta_{\text{end}}^{1/2} - \beta_{\text{start}}^{1/2})\right)^2, \tag{7}$$

Sigmoid schedules, characterized by an S-shaped curve (e.g., derived from a logistic function), provide a smooth transition with slow noise growth at the extremes and a rapid increase in the middle:

$$\beta_t = \beta_{\text{start}} + (\beta_{\text{end}} - \beta_{\text{start}})\cdot\frac{1}{1+e^{-\left(-6+\frac{12(t-1)}{T-1}\right)}}, \tag{8}$$

III. Multifractal textures

Modeling. Following Mandelbrot's seminal work [14], Multifractal random walks (MRW) [15] have become widely used as versatile models for real-world textures characterized by scale-free statistics [12], [16]. MRW, $X(\underline{r})$, are classically defined by means of two independent zero-mean Gaussian processes, $G_H(\underline{r})$ and $\omega_\lambda(\underline{r})$, as:

$$X(\underline{r}) = G_H(\underline{r})e^{\omega_\lambda(\underline{r})}. \tag{9}$$

The process $G_H(\underline{r})$ consists of the well-known 2D fractional Gaussian noise (2D-fGn), a reference Gaussian model for scale-free textures, fully defined by its covariance function, controlled by the Hurst exponent H [17], [18]. The process $\omega_\lambda(\underline{r})$ is defined via its covariance function, designed to induce multifractality in spatial statistics:

$$C_{MF}\underline{r} = \lambda^2\log\left(\frac{L}{\|\underline{r}\|+1}\right), \tag{10}$$

(for $\|\underline{r}\| \leq L$ and 0 otherwise, with L as an arbitrary integral scale). It is thus fully controlled by the multifractality parameter λ [15].

Analysis. Let $\{d_X(j,\underline{k})\}$, denote the discrete wavelet transform coefficients of texture X, defined as inner products between X and dilated (at scale 2^j) and translated (at location $2^j\underline{k}$) templates based on a tensor-product 2D wavelet [19]. Let us further define wavelet leaders $L(j,\underline{k})$ as local suprema of wavelet coefficients, taken over finer scales and within a short spatial neighborhood [20], [21].

It has been well-documented (cf. e.g., [20]) that, for MRW, the first-order cumulant $C_1(2^j)$ and second-order cumulant $C_2(2^j)$ of $\ln L_X$ behave linearly in $\ln 2^j$:

$$C_1(2^j) = c_1^{(0)} + c_1\ln 2^j, \quad C_2(2^j) = c_2^{(0)} + c_2\ln 2^j, \tag{11}$$

with $c_1 = H - \lambda^2/2$ and $c_2 = -\lambda^2$.

In essence, the behavior of $C_1(2^j)$ across scales is dominated by $H \gg \lambda^2$ and primarily describes the second-order statistics of X, thereby capturing its global correlation structure. Conversely, the behavior of $C_2(2^j)$ across scales, controlled by λ, characterizes the multifractality of the texture, i.e., the tight organization across space of local transient structures, not accounted for by the correlation function.

These wavelet and wavelet-leader multiscale statistics, $C_1(2^j)$ and $C_2(2^j)$, will be used here to assess the quality of the diffusive network synthesized textures[2].

[2]They are implemented using the documented toolbox available at www.irit.fr/ Herwig.Wendt/software.html.

It is important to note that multifractal characterization is used solely to validate the quality of texture synthesis and not for constructing the loss function used to train the diffusion network. Thus, multifractal and scale-free properties are never directly utilized to inform the diffusion network.

IV. Experimental Methodology

A. Training dataset

The training dataset comprises 1000 independent samples of homogeneous univariate multifractal random walk (MRW) textures, each of size 256×256, synthesized to exhibit scale-free statistics and multifractal properties as defined in Section III. These are generated using Matlab routines implementing circulant matrix embedding [4], [22], with fixed parameters: Hurst exponent $H = 0.8$, and multifractality parameter $\lambda^2 = 0.01$, set uniformly across samples for consistent statistical characteristics.

B. Model Training

The diffusion model adheres to the classical DDPM framework proposed by Ho et al. [3], utilizing 1000 timesteps and a U-Net architecture for denoising. Training minimizes the mean-squared error (MSE) loss between the predicted noise $\epsilon_\theta(\mathbf{x}_t, t)$ and actual noise ϵ, optimizing the reverse process to reconstruct textures from noise. The model employs a U-Net architecture inspired by ConvNeXt blocks to perform the denoising process [23]. The U-Net follows an encoder-decoder structure with skip connections, processing 256×256 images through progressive downsampling and upsampling stages. The usage of ConvNeXt blocks enhances efficiency by incorporating depthwise-separable convolutions and residual connections with self-attention layers at lower resolutions to capture long-range dependencies [24]. Sinusoidal positional embeddings encode the diffusion timestep t, enabling adaptive denoising [25]. Skip connections concatenate encoder and decoder features, preserving fine details, and a final 1×1 convolution produces the denoised output.

Training was conducted on an AD102GL [RTX 6000 Ada Generation] GPU. We test four noise schedulers—linear, cosine, quadratic, and sigmoid (Section II)—across 2000 epochs with Adam (learning rate 0.0002, batch size 32). Each scheduler was trained independently to synthesize multifractal properties.

C. Evaluation Metrics

To evaluate the quality of textures synthesized by the diffusion model, we employ multifractal analysis metrics defined in Section III. Specifically, we compute the first and second cumulants of the logarithm of wavelet leaders, $C_1(j)$ and $C_2(j)$, across scales 2^j, for both the original training dataset (1000 MRW textures with $H = 0.8$, $\lambda^2 = 0.01$) and the generated textures. We assess the fidelity of these scaling properties through two complementary approaches:

1) We plot the averaged $C_1(j)$ and $C_2(j)$ curves for both the original and generated textures across scales. These plots allow us to qualitatively assess whether the scaling behavior of the generated textures aligns with that of the original dataset.
2) To provide a quantitative measure of the multifractal properties, we estimate the slopes $\hat{c}_1$ and $\hat{c}_2$ from linear regressions of $C_1(j)$ and $C_2(j)$ across scales $2^{j_1} = 2$ to $2^{j_1} = 4$, respectively. We then calculate the mean and standard deviation of these estimated slopes across 1000 samples for both the original and generated textures. These statistics allow us to assess how well the generated textures reproduce the multifractal scaling properties of the original dataset.

V. Results and Discussion

This section presents the outcomes of synthesizing multifractal textures using the diffusion model. Figure 1 displays representative texture samples: (a) a training sample, and generated samples for (b) linear, (c) cosine, (d) quadratic, and (e) sigmoid schedulers. Visually, the linear, cosine, and sigmoid schedulers produce textures that are visually similar to that of the training sample. Conversely, the quadratic scheduler yields visually very different textures, indicating potential challenges in preserving texture coherence.

Figure 2 reports the functions $C_1(j)$ and $C_2(j)$ as functions of scales for the training (blue) and diffusion network generated sets, with theoretical slopes (dashed black) at $c_1 = 0.795$ and $c_2 = -0.01$. These functions are obtained as averages across 1000 sample textures, also yielding confidence intervals. To enhance readability, the functions $C_1(j)$ and $C_2(j)$ for each scheduler are shifted by 0.05 on the x-axis.

Table I summarizes the means and standard deviations of $\hat{c}_1$ and $\hat{c}_2$, estimated by linear regressions across scales $2^{j_1} = 2$ to $2^{j_1} = 4$, averaged over 1000 samples, for the training data and for diffusion model textures generated by different schedulers.

For $C_1(j)$, the training data starts around -1.55 at scale 2^1 and increases steadily to 1.61 by scale 2^5, with a slope closely aligning with the theoretical value of 0.795 (dashed black line). The linear scheduler (red) begins at -1.51 and rises to 1.58, with estimated $c_1^{linear} = 0.799$, closely reproducing the function $C_1(2^j)$ from the training set, with small standard deviations, indicating consistent synthesis. Moreover, the standard deviations for the linear scheduler are slightly smaller but comparable in magnitude to those of the training set. Interestingly, this shows that the linear noise scheduler not only reproduces the global scaling property on average but also in distribution, i.e., in variability across samples.

The sigmoid (green) and cosine (yellow) schedulers start at -1.73 and -1.61 respectively, rising more gradually to 1.90 and 1.84 by scale 2^5, suggesting a slight overestimation of the scaling behavior compared to the training data, $\hat{c}_1^{sigmoid} = 0.94$ and $\hat{c}_1^{cosine} = 0.87$. The quadratic scheduler (cyan) starts at -1.5391 and increases more steeply to 1.9128 by scale 2^5, yielding $\hat{c}_1^{quadratic} = 0.92$, with larger standard deviations, indicating higher variability and potential instability.

For $C_2(j)$, the training data yields a decline across scales with slope $\hat{c}_2^{training} = -0.010$ very much in agreement with the theoretical $c_2 = -0.01$.

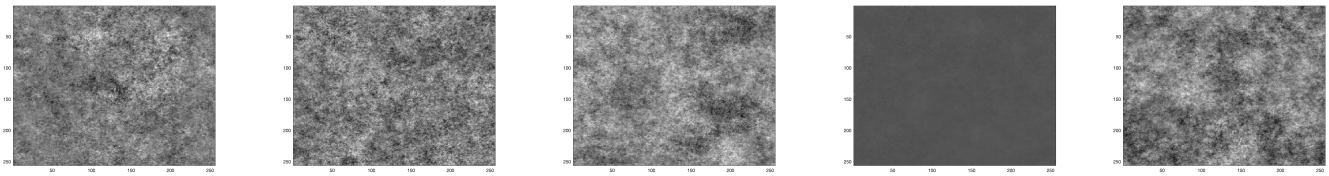

Fig. 1: **Texture samples.** (a) Sample from the training dataset. Diffusion model generated samples, using different noise schedulers: (b) Linear, (c) Cosine, (d) Quadratic, (e) Sigmoid.

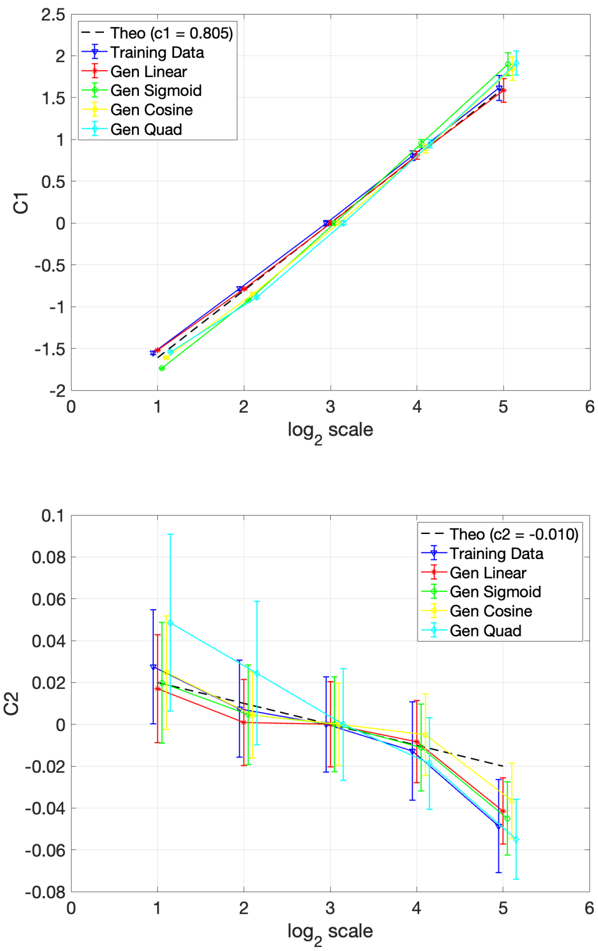

Fig. 2: **Cumulants $C_1(2^j)$ and $C_2(2^j)$ as functions of scales**, averaged across the training set (blue) and diffusion model generated texture sets, obtained from different noise schedulers. Top: $C_1(2^j)$. Bottom: $C_2(2^j)$. The dashed black line materializes the theoretical scaling behavior across scales.

The linear scheduler produces a function $C_2(j)$ that decreases across scale with a slope $\hat{c}_2^{Linear} = -0.004$, which indicates that multifractality is actually induced in the diffusion model generated textures, yet with a slightly lower intensity compared to the expected one or to the one estimated from the training set. Notably, the standard estimation across diffusion model generated samples is of the same order of magnitude as that obtained from the training set, showing that diffusion models not only reproduce the multifractality parameter on average, but also in variability across samples.

Interestingly, the other noise schedulers introduce multifractality in textures, with the sigmoid scheduler ($\hat{c}_2 = -0.0066$) closely matching the training slope of -0.010, while the cosine ($\hat{c}_2 = -0.0036$) and quadratic ($\hat{c}_2 = -0.0209$) schedulers underestimate and overestimate it, respectively. Their standard deviations (0.0095, 0.0088, 0.0103) remain similar to the training set's 0.0119.

The variations in the reproduction of $C_1(j)$ across schedulers highlight the nuanced impact of noise scheduling strategies on texture synthesis outcomes. Specifically, the linear scheduler's fidelity to the training data and its low variability suggest that it is well-suited for applications requiring precise preservation of global scaling properties. In contrast, the other schedulers overestimate the scaling behavior, making them less suitable for such applications.

Furthermore, the variations in the reproduction of $C_2(j)$ across schedulers underscore the challenges of capturing multifractality through noise scheduling strategies in texture synthesis. While the linear scheduler struggles to fully match the training data's multifractal intensity, it nonetheless demonstrates a more balanced performance across both average multifractality and variability. As a result, the linear scheduler emerges as the better choice on average for applications requiring controlled multifractal texture synthesis.

TABLE I: Means and standard deviations of $\hat{c}_1$ and $\hat{c}_2$, averaged over 1000 realizations for the training and for the diffusion model generated texture sets.

Dataset	$\hat{c}_1$		$\hat{c}_2$	
	Mean	**Std**	**Mean**	**Std**
Training	0.7950	0.0394	-0.0101	0.0119
Sigmoid	0.9378	0.0351	-0.0066	0.0095
Linear	0.7994	0.0335	-0.0036	0.0093
Cosine	0.8686	0.0364	-0.0036	0.0088
Quadratic	0.9166	0.0341	-0.0209	0.0103

VI. Conclusion

This study explored the capability of diffusion neural networks to synthesize multifractal textures, leveraging a U-Net-based model trained on 1000 univariate MRW textures under four noise schedulers—linear, cosine, quadratic, and sigmoid. We targeted textures with complex, scale-invariant dynamics, assessed via wavelet-leader multifractal statistics ($C_1(j)$, $C_2(j)$) and their moments, without embedding such properties in the training loss. among the schedulers tested, the linear scheduler demonstrated the best performance, both correctly reproducing global correlations ($C_1(2^j)$), and generating multifractality ($C_2(2^j)$), yet so far not perfectly. These findings highlight the potential of diffusion models for synthesizing texture, yet suggest room for improvement. Future work will extend this approach to multivariate multifractal textures or to textures combining multifractal properties with anisotropy and design specialized diffusion architectures to better handle their intricate statistical structures.

ACKNOWLEDGMENT

We sincerely thank Pulsalys for supporting our work. We also acknowledge the support of the Centre Blaise Pascal's IT test platform at ENS de Lyon (Lyon, France) for Machine Learning facilities. The platform operates the SIDUS solution [26] developed by Emmanuel Quemener.

REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[2] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

[3] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[4] P. Abry, V. Mauduit, E. Quemener, and S. Roux, "Multivariate multifractal texture dcgan synthesis: How well does it work? how does one know?," *Journal of Signal Processing Systems*, pp. 1–17, 2022.

[5] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International conference on machine learning*, pp. 2256–2265, pmlr, 2015.

[6] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," *arXiv preprint arXiv:2011.13456*, 2020.

[7] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in neural information processing systems*, vol. 34, pp. 8780–8794, 2021.

[8] D. Z. Chen, Y. Siddiqui, H.-Y. Lee, S. Tulyakov, and M. Nießner, "Text2tex: Text-driven texture synthesis via diffusion models," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 18558–18568, 2023.

[9] T. Cao, K. Kreis, S. Fidler, N. Sharp, and K. Yin, "Texfusion: Synthesizing 3d textures with text-guided image diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4169–4181, 2023.

[10] X. Yu, Z. Yuan, Y.-C. Guo, Y.-T. Liu, J. Liu, Y. Li, Y.-P. Cao, D. Liang, and X. Qi, "Texgen: a generative diffusion model for mesh textures," *ACM Transactions on Graphics (TOG)*, vol. 43, no. 6, pp. 1–14, 2024.

[11] F. J. Richard, "Anisotropy of hölder gaussian random fields: characterization, estimation, and application to image textures," *Statistics and Computing*, vol. 28, pp. 1155–1168, 2018.

[12] S. Jaffard, P. Abry, and H. Wendt, *Irregularities and Scaling in Signal and Image Processing: Multifractal Analysis*, pp. 31–116. Singapore: World scientific publishing, 2015.

[13] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *International conference on machine learning*, pp. 8162–8171, PMLR, 2021.

[14] 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.

[15] E. Bacry, J. Delour, and J.-F. Muzy, "Multifractal random walk," *Phys. Rev. E*, vol. 64: 026103, 2001.

[16] C. Meneveau, K. Sreenivasan, P. Kailasnath, and M. Fan, "Joint multifractal measures - theory and applications to turbulence," *Physical Review A*, vol. 41, pp. 894–913, January 1990.

[17] G. Samorodnitsky and M. Taqqu, *Stable non-Gaussian random processes*. New York: Chapman and Hall, 1994.

[18] S. G. Roux, M. Clausel, B. Vedel, S. Jaffard, and P. Abry, "Self-similar anisotropic texture analysis: The hyperbolic wavelet transform contribution," *IEEE Transactions on Image Processing*, vol. 22, no. 11, pp. 4353–4363, 2013.

[19] S. Mallat, *A Wavelet Tour of Signal Processing*. San Diego, CA: Academic Press, 1998.

[20] H. Wendt, P. Abry, and S. Jaffard, "Bootstrap for empirical multifractal analysis," *IEEE Signal Proc. Mag.*, vol. 24, no. 4, pp. 38–48, 2007.

[21] H. Wendt, S. G. Roux, P. Abry, and S. Jaffard, "Wavelet leaders and bootstrap for multifractal analysis of images," *Signal Proces.*, vol. 89, no. 6, pp. 1100–1114, 2009.

[22] 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.

[23] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11976–11986, 2022.

[24] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," *arXiv preprint arXiv:1803.02155*, 2018.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[26] E. Quemener and M. Corvellec, "Sidus—the solution for extreme deduplication of an operating system," *Linux J.*, vol. 2013, Nov. 2013.