Improved Blue Noise for Diffusion Models

Shaked Caspi and Guy Levy

Tel-Aviv University, Israel.

Contributing authors: shakedcaspi@mail.tau.ac.il;
guylevy3@mail.tau.ac.il;

Abstract

Diffusion-based generative models have revolutionized image synthesis, surpassing traditional approaches such as Generative Adversarial Networks (GANs) in terms of sample quality, stability, and diversity. These models, which iteratively refine noisy inputs through a learned denoising process, have achieved state-of-the-art performance in various applications, including image generation, inpainting, and super-resolution [1, 2]. Prominent examples such as Stable Diffusion [3], ADM [4], and Consistency Models [5] rely on uncorrelated Gaussian noise during both training and sampling. Recent research suggests that structured noise patterns, such as blue noise, can further enhance generative quality by influencing the frequency characteristics of synthesized images [6]. In this work, we analyze the impact of different correlated noise distributions on a pretrained diffusion model. Specifically, we introduce and evaluate the effects of injecting various structured noise types—including blue noise, red noise, and other correlated patterns, in the early stages of the generative process. By modifying the noise generation strategy, we aim to better regulate the frequency components learned by the model, potentially improving perceptual quality and generative fidelity. We perform experiments on the AFHQ-Cat dataset, comparing multiple noise configurations and assessing their impact using standard evaluation metrics such as Fréchet Inception Distance (FID) [7]. Our best model outperformed Blue Noise for Diffusion Models (BNDM) by Huang et al. [6], achieving an FID of 6.9 vs. 8.96. Additionally, our noise generation is much faster (seconds) while maintaining similar quality.

Keywords: Diffusion Models, Generative Modeling, Noise Injection, Correlated, Image Generation, Optimization in Diffusion Models

1 Introduction

Diffusion models have emerged as a powerful class of generative models, surpassing traditional approaches such as Generative Adversarial Networks (GANs) in terms of image quality and training stability [1, 2]. These models operate by gradually adding Gaussian noise to input data in a forward process, followed by a learned denoising process that reconstructs the original distribution. This iterative refinement mechanism enables high-quality synthesis across a range of generative tasks. However, most existing diffusion models rely exclusively on uncorrelated Gaussian noise, which may not optimally account for the frequency content reconstructed by the denoising network [6, 8]. This raises the question of whether alternative noise distributions, particularly structured and correlated noise, could enhance generative performance by influencing the model's learning dynamics and convergence properties.
A recent study, "Blue Noise for Diffusion Models" by Xingchang Huang et al. [6], explored the use of blue noise as a replacement for standard Gaussian noise in diffusion models. Blue noise, characterized by its reduced low-frequency components, has been widely used in graphics applications for perceptual quality enhancement [9]. The authors demonstrate that injecting blue noise into the diffusion process improves sample diversity and reduces unwanted low-frequency artifacts in generated images. Their findings suggest that noise characteristics play a fundamental role in shaping diffusion model performance, motivating further exploration into structured noise patterns.
In this work, we extend this investigation by analyzing the impact of different structured noise distributions on the IADB model as our baseline model. We introduce structured noise patterns—including blue noise, red noise, purple noise, and other frequency-selective noise.
Our experimental results assess the effects of structured noise injection using Fréchet Inception Distance (FID) [7], PRDC (precision, recall, diversity and coverage) as evaluation metrics. Our best model outperformed the one presented in Blue Noise for Diffusion Models (BNDM) by Huang et al. [6], achieving an FID of 6.9 compared to 8.96. Additionally, our noise generation technique is significantly faster, producing noise within seconds, while the approach proposed in BNDM [6] requires considerably more time, despite yielding similar quality results.

2 Related Work

Recent advancements in diffusion models have explored the integration of various noise types to enhance generative performance and address specific challenges in image synthesis. Notably, the study "Blue Noise for Diffusion Models" by Xingchang Huang et al. [6] investigates the incorporation of blue noise—a noise pattern characterized by the absence of low-frequency components—into the diffusion process. The authors demonstrate that using blue noise instead of standard Gaussian noise can improve sample diversity and reduce low-frequency artifacts, thereby enhancing the perceptual quality of

generated images.
Another work is "Removing Structured Noise with Diffusion Models" by Tristan S. W. Stevens et al. [10]. This study extends diffusion models to handle structured noise in inverse problems such as denoising and compressed sensing. The authors propose a joint conditional reverse diffusion process with learned scores for both noise and signal-generating distributions, demonstrating a performance gains over competitive baselines.
Additionally, the paper "Diffusion Models with Learned Adaptive Noise" by Subham Sekhar Sahoo et al. [11] introduces a method where the diffusion process learns adaptive noise schedules from data. This approach allows the model to apply noise at different rates across an image, leading to improved log-likelihood estimation and sample quality.

3 Data

For our experiments, we used the AFHQ-Cat dataset [12] at a resolution of 64×64, which was also utilized in the baseline study "Blue Noise for Diffusion Models" by Xingchang Huang et al. [6] This dataset is a subset of AFHQ (Animal Faces-HQ), a high-quality image collection commonly used for generative modeling tasks. The AFHQ-Cat dataset consists of 5153 images of cats with diverse appearances, making it suitable for evaluating the impact of different noise injection strategies on image generation [12].
To ensure a valid and direct comparison with previous work, we downloaded the dataset directly from the original paper's GitHub repository without any modifications or preprocessing. This approach guarantees consistency in the experimental setup and eliminates potential discrepancies arising from data variations.

4 Methods

4.1 Type of Noises

Before detailing our methods, we provide a brief overview of noise and its various types. White noise is modeled as an independent and identically distributed (i.i.d.) random vector drawn from a standard distribution. It is referred to as "white" noise because its Fourier transform exhibits a uniform power spectral density (PSD). White noise plays a crucial role in diffusion models, particularly during the forward diffusion process, where noise is incrementally added to a real image to facilitate the learning of a denoising function through the loss function.
In this work, we extend the study presented in Blue Noise for Diffusion Models by Huang et al. [6]. In that paper, the authors trained a diffusion model, IADB [8], by interpolating blue and red noise with white noise, rather than exclusively using white noise as is typically done in diffusion models, including in the original IADB paper. Different types of noise correspond to distinct color maps, each defined by unique frequency distributions. For example, blue

noise exhibits greater power at higher frequencies, with power decreasing as the frequency decreases.

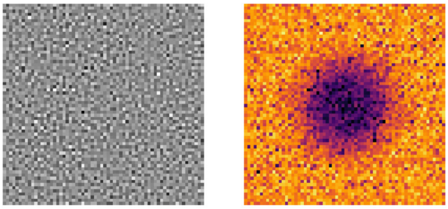

Fig. 1 64×64 gray scale blue noise (left); PSD of blue noise (right)

4.2 Produce a Noise With a Specific PSD

Several algorithms exist for generating structured noise, each employing different optimization strategies to shape the frequency distribution. Generating correlated noise in a high dimension is computationally expensive, so the sampling task is not trivial. In "Blue Noise for Diffusion Models" the authors employed simulated annealing using objective function presented by Ulichney, [9] to generate multiple blue noise masks. This approach requires significant time to process [6]. Once we sample enough masks we can calculate the covariance matrix Σ. Then applying Cholesky decomposition ($\Sigma = LL^T$) to extract the triangular matrix L. Finally, the matrix L is multiplied by $\epsilon \sim N(0,1)$ to generate a new blue noise mask. In our work, we explored three algorithms to

Fig. 2 Our L matrix that generates blue noise

sample correlated noise simulated, annealing, direct amplitude swapping and direct gradient descent optimization.

4.2.1 Simulated Annealing

Simulated annealing is a probabilistic optimization technique. The process works by randomly swapping the values of two pixels in an image and then calculating the objective function. If the loss decreases after the swap, the change is accepted. Otherwise, the swap may still be accepted with a certain probability, which decreases over time, allowing the algorithm to escape local minima and explore better solutions. The main disadvantage of this algorithm is that it takes long time to produce good results.

4.2.2 Direct Amplitude Swapping

Direct amplitude swapping is a technique for modifying noise by working in the Fourier domain. It starts by computing the Fourier transform of the noise and splitting it into phase and amplitude components. The amplitude is then replaced with a target function that matches the desired noise characteristics. For example, for blue noise, the target amplitude is proportional to f^2. Finally, the modified amplitude is combined with the original phase, and the inverse FFT is applied to reconstruct the noise. This technique isn't ideal, as it tends to generate noise patterns that lack diversity.

4.2.3 Gradient Descent Optimization

This optimization process is straightforward. We define the loss function as follows:

$$\|P(x) - P_{\text{target}}\|^2$$

where $P(x)$ is the power spectrum and P_{target} is the desired power spectrum. We then apply an iterative gradient descent process until convergence or until reaching the maximum number of iterations. For blue noise, P_{target} is defined as:

$$P_{\text{target}} = \left(\frac{f}{\text{threshold}}\right)^{\text{exp}} \cdot \mathbb{I}\{f < \text{threshold}\}$$

where $\exp > 1$ and threshold $\in [0, 1]$. These parameters are manually controlled by us.

After experimenting with the three algorithms, we decided to proceed with the latest one for two main reasons. First, this algorithm is significantly faster than the other two, allowing us to generate 100k masks per noise efficiently. Additionally, it makes it easy to produce diverse noise patterns.

5 Experiments

5.1 Implementation Details

We used AFHQ-Cat [12], with 64×64 resolution, for unconditional image generation. Our framework is implemented in Pytorch [13] based on the official implementations from Song et al. [2] and Heitz et al. [8] We used 2D U-Net [14] implemented in diffusers library [15]. Regarding the hyperparameters in the IADB model, we used $T = 1000$ for training and $T = 250$ testing. To optimize the network parameters, we used AdamW optimizer [16] with learning rate

0.0001. We used 1 NVIDIA GeForce RTX 4070 Super (12GB) GPU to train and test.
For evaluation, we generated **5153** (entire AFHQ-Cat dataset size) images from each model using the same seed, used FID [7], and PRDC (Precision, Recall, Diversity, Coverage) [17] to measure the generative quality of all models with Inception-v3 network [18] as backbone.

5.2 Training Details

We utilized the IADB model extension along with the corresponding weights provided by Huang et al. in "Blue Noise for Diffusion Models" [6]. The model was then fine-tuned for an additional 20 epochs, with the batch size reduced to 32 due to memory constraints.
For each training iteration, we used this pre-trained model with a newly computed covariance matrix, obtained through the gradient descent optimization process described earlier.
Subsequently, we selected the best structural noise model and trained it from scratch for 1,000 epochs. We saved the model weights every 50 epochs to assess performance at different stopping points, allowing us to mitigate potential overfitting or underfitting.

5.3 Results

First, we experimented with 16 different noise patterns (see Appendix) and selected the top 5 based on FID and PRDC scores. Additionally, we created a baseline model using the original noise-covariance matrix provided by Huang et al. [6], trained under the same settings described above. The overall best model was "bluev21", with 12.11 FID.

5.3.1 The noises that we have tested:

1. **Red**: characterized by low power at high frequencies and vice versa.
2. **Green**: characterized by high power at only medium frequencies.
3. **Blue**: Characterized by high power at high frequencies that gradually decreases as the frequencies decrease, we tested 25 different variations of blue noise, each differing slightly in its behavior at low and medium frequencies. The image presents the two best-performing types of blue noise.
4. **Purple**: Characterized by high power at high frequencies that gradually decreases as the frequencies decrease, similar to blue noise but stronger.
5. **White**: simply white noise.
6. **Baseline**: blue noise from "Blue Noise for Diffusion Models"

Fig. 3 The PSD and generated images of top 5 noises and the baseline

Table 1 Different noise based models performances using pre-trained IADB with additional 20 epochs

Model Name	FID	Precision	Recall	Density	Coverage
Red (ours)	**11.6406**	0.7153	0.4974	0.7279	0.7301
Bluev22 (ours)	13.2082	**0.7789**	0.4582	**0.9353**	0.7737
Bluev21 (ours)	12.1196	0.76	**0.5319**	0.8808	**0.7794**
White (default)	17.9401	0.7046	0.4694	0.6945	0.6451
Baseline (theirs)	13.7027	0.7557	0.4989	0.7886	0.7132

Our next experiment involved selecting the best overall noise, referred to as "bluev21", and training the entire IADB model from scratch using the covariance matrix "bluev21". We evaluated the model at epochs 500, 750, and 1000, with the best performance observed at 750 epochs, achieving an FID score of 6.9. This model significantly outperforms the original model presented in "Blue Noise for Diffusion Models" by Xingchang Huang et al. [6].

Table 2 Comparison of different training configurations from scratch, evaluating FID, Precision, Recall, Density, and Coverage.

Model Name	FID	Precision	Recall	Density	Coverage
Baseline (theirs)	8.9628	0.8151	0.6061	1.1147	0.8700
bluev21-500 (ours)	8.7107	0.7423	0.6319	0.8848	0.8368
bluev21-750 (ours)	**6.9013**	**0.8494**	0.6006	**1.2833**	**0.9156**
bluev21-900 (ours)	8.4734	0.7586	0.6303	0.9499	0.8874
bluev21-1000 (ours)	9.1822	0.7224	**0.6369**	0.7921	0.8253

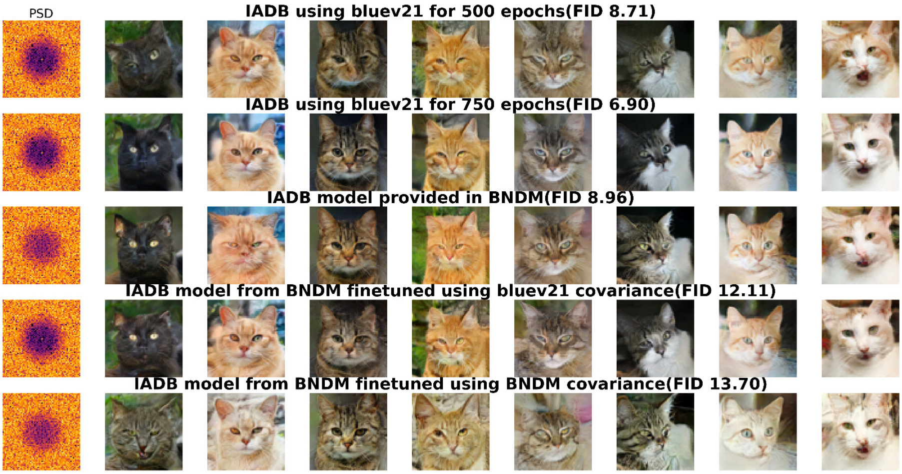

Fig. 4 Comparison of different generated cats by "bluev21", BNDM and BNDM fine-tuned models

6 Conclusions

Our primary finding is that we significantly outperformed the original model presented in "Blue Noise for Diffusion Models" by Xingchang Huang et al. [6] across all performance metrics in BNDM, including FID, Precision, and Recall. This result highlights a promising direction for further research, demonstrating the potential of incorporating correlated noise into the diffusion process rather than relying solely on Gaussian noise.

Another key contribution of our work is the method used to generate noise patterns. We introduce a simple yet efficient optimization process that enables the generation of a vast number of noise masks (100,000) within seconds. This represents a major improvement over the original approach, which relied on simulated annealing—an accurate but computationally expensive method.

Furthermore, our technique has the potential to be integrated directly into the training process, optimizing noise in real time rather than as a separate pre-processing step. This opens new possibilities for future research in enhancing diffusion-based models.

Appendix A Source Code

Our code [19] is based on the open source repository BNDM [6].

Appendix B Full Results

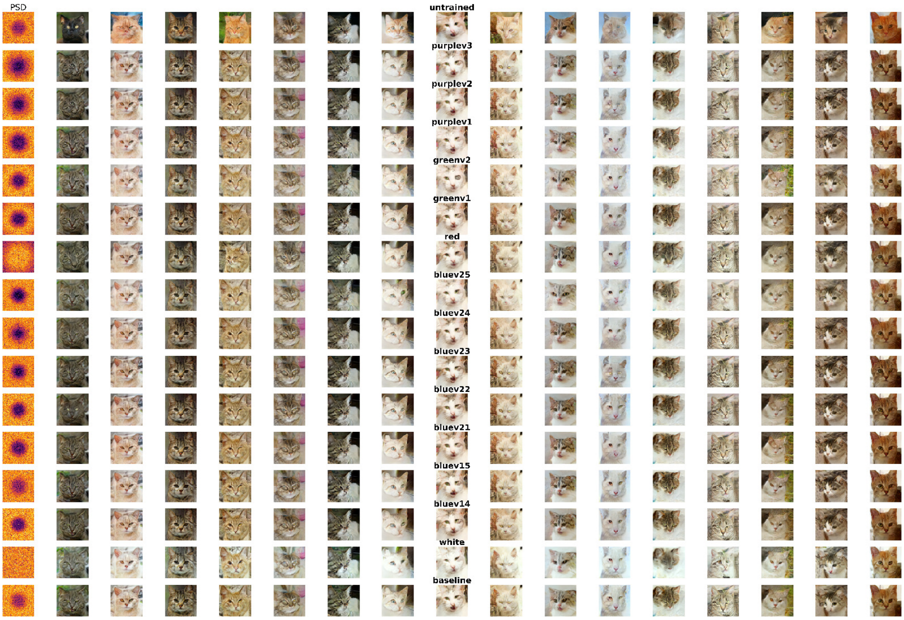

Fig. B1 Results of all noises that we have tested

References

[1] Ho, J., Jain, A., Abbeel, P.: Denoising Diffusion Probabilistic Models (2020). https://doi.org/10.48550/arXiv.2006.11239. https://arxiv.org/abs/2006.11239

[2] Song, J., Meng, C., Ermon, S.: Denoising Diffusion Implicit Models (2022). https://arxiv.org/abs/2010.02502

[3] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-Resolution Image Synthesis with Latent Diffusion Models (2022). https://arxiv.org/abs/2112.10752

[4] Dhariwal, P., Nichol, A.: Diffusion Models Beat GANs on Image Synthesis (2021). https://arxiv.org/abs/2105.05233

[5] Song, Y., Dhariwal, P., Chen, M., Sutskever, I.: Consistency Models (2023). https://arxiv.org/abs/2303.01469

[6] Huang, X., Salaün, C., Vasconcelos, C., Theobalt, C., Öztireli, C., Singh, G.: Blue noise for diffusion models (2024). https://arxiv.org/abs/2402.04930

[7] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium (2018). https://arxiv.org/abs/1706.08500

[8] Heitz, E., Belcour, L., Chambon, T.: Iterative -(de)Blending: a Minimalist Deterministic Diffusion Model. ACM (2023). https://doi.org/10.1145/3588432.3591540. http://dx.doi.org/10.1145/3588432.3591540

[9] Ulichney, R.: The void-and-cluster method for dither array generation (1993). https://cv.ulichney.com/papers/1993-void-cluster.pdf

[10] Stevens, T.S.W., van Gorp, H., Meral, F.C., Shin, J., Yu, J., Robert, J.-L., van Sloun, R.J.G.: Removing Structured Noise with Diffusion Models (2023). https://arxiv.org/abs/2302.05290

[11] Sahoo, S.S., Gokaslan, A., Sa, C.D., Kuleshov, V.: Diffusion Models With Learned Adaptive Noise (2024). https://arxiv.org/abs/2312.13236

[12] Choi, Y., Uh, Y., Yoo, J., Ha, J.-W.: StarGAN v2: Diverse Image Synthesis for Multiple Domains (2020). https://arxiv.org/abs/1912.01865

[13] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: PyTorch: An Imperative Style, High-Performance Deep Learning Library (2019)

[14] Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation (2015). https://arxiv.org/abs/1505.04597

[15] von Platen, P., Patil, S., Lozhkov, A., Cuenca, P., Lambert, N., Rasul, K., Davaadorj, M., Nair, D., Paul, S., Berman, W., Xu, Y., Liu, S., Wolf, T.: Diffusers: State-of-the-art diffusion models. GitHub (2022)

[16] Loshchilov, I., Hutter, F.: Decoupled Weight Decay Regularization (2019). https://arxiv.org/abs/1711.05101

[17] Naeem, M.F., Oh, S.J., Uh, Y., Choi, Y., Yoo, J.: Reliable Fidelity and Diversity Metrics for Generative Models (2020). https://arxiv.org/abs/2002.09797

[18] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the Inception Architecture for Computer Vision (2015). https://arxiv.org/abs/1512.00567

[19] Guy Levy, S.C.: Improved-Blue-Noise-for-Diffusion-Models (2025). https://github.com/No-Guy/Improved-Blue-Noise-for-Diffusion-Models.git