Denoising Diffusion Probabilistic Model for Realistic Financial Correlation Matrices

Szymon Kubiak
City, University of London
UK
szymon.kubiak@city.ac.uk

Tillman Weyde
City, University of London
UK
t.e.weyde@city.ac.uk

Oleksandr Galkin
City, University of London
UK
Oleksandr.Galkin@city.ac.uk

Daniel Philps
University of Warwick
UK
Daniel.philps@wbs.ac.uk

Ram Gopal
Warwick Business School
UK
ram.gopal@wbs.ac.uk

Abstract

Financial correlation matrices play a vital role in various quantitative finance applications, but generating synthetic correlation matrices that accurately reflect market structures and stylized facts remains challenging. We introduce a novel application of denoising diffusion probabilistic models (DDPMs) for this task, proposing both unconditional (DM) and conditional (CDM) models. Our experimental evaluation demonstrates the superior performance of our models in generating correlation matrices that closely resemble empirical data while capturing differences across market regimes. We also present a case study highlighting the utility of our approach in assessing asset allocation frameworks and enhancing risk modeling by augmenting empirical datasets with synthetic data. Our findings showcase DDPMs' potential in mitigating limitations of scarce financial data, enabling robust quantitative modeling and analysis.

Keywords

Generative machine learning models, synthetic data, correlation matrices, asset allocation, financial risk management

ACM Reference Format:
Szymon Kubiak, Tillman Weyde, Oleksandr Galkin, Daniel Philps, and Ram Gopal. 2024. Denoising Diffusion Probabilistic Model for Realistic Financial Correlation Matrices. In *5th ACM International Conference on AI in Finance (ICAIF '24), November 14–17, 2024, Brooklyn, NY, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3677052.3698640

1 Introduction

Financial correlation matrices capture the co-movement patterns among financial assets. They play a crucial role in various quantitative finance applications, such as portfolio construction, risk modeling, complex derivative pricing, and asset allocation. However, generating realistic synthetic correlation matrices that accurately capture underlying market structures and stylized facts remains a significant challenge [15, 25].

Recent developments in machine learning have led researchers to explore novel techniques for sampling correlation matrices. Notable applications include Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), which have shown promise in producing synthetic correlation matrices that resemble their empirical counterparts [1, 20, 25, 26]. However, the exploration of machine learning in generating more realistic correlation matrices remains an open field.

Denoising diffusion probabilistic models (DDPMs) have recently emerged as a powerful tool for generating high-quality synthetic data, showcasing remarkable performance in various computer vision tasks [4, 14, 29, 30]. Their success in image generation motivates us to investigate their potential in outperforming existing machine learning architectures for generating realistic financial correlation matrices.

This study explores the application of DDPMs to generate synthetic financial correlation matrices. We present both unconditional and conditional models that replicate diverse market structures across various regimes. Our approach provides quantitative finance practitioners with a method to augment limited historical data. By generating realistic synthetic data, investors and risk managers can analyze portfolio performance and risk across a wider range of market conditions, potentially leading to more resilient investment strategies and improved decision-making processes.

The main contributions of this paper are as follows:

- We introduce a novel application of DDPMs for generating realistic financial correlation matrices.
- We propose both unconditional and conditional DDPMs, incorporating market conditions through continuous volatility-based conditioning variables.
- We demonstrate the superior performance of our models compared to existing methods through extensive experiments on diverse datasets.
- We present a case study showcasing the practical value of our approach in enhancing financial modeling.

2 Related Work

2.1 Sampling Financial Correlation Matrices

Financial correlation matrices exhibit several key stylized facts [25], including a significantly positive shift in the distribution of

ICAIF '24, November 14–17, 2024, Brooklyn, NY, USA

ACM ISBN 979-8-4007-1081-0/24/11
https://doi.org/10.1145/3677052.3698640

pairwise correlations, eigenvalues following the Marchenko-Pastur distribution with a large first eigenvalue (market) and a few other large eigenvalues (industries), the Perron-Frobenius property, a hierarchical structure of correlations, and a scale-free property of the corresponding Minimum Spanning Tree. However, developing models that effectively generate correlation matrices adhering to all of these characteristics has proven challenging [15].

Traditional approaches for generating synthetic correlation matrices, such as the onion method [7], vine copulas [21], and matrix perturbation techniques [11], often struggle to replicate all these stylized facts and fail to produce samples that accurately mirror real-world financial correlation structures.

Recent literature has proposed innovative machine learning techniques to sample realistic correlation matrices. One such approach is CorrGAN [25], an application of Generative Adversarial Networks (GANs) [8] based on the DCGAN architecture [28]. [20] proposed a modified version based on Wasserstein GAN [9], and [26] extended the model to condition sampled data on market regimes. In parallel, [1] leveraged Variational Autoencoders (VAEs) [19] to sample correlation matrices effectively.

An alternative approach involves generating correlation matrices by sampling synthetic financial time series. This can be achieved through traditional methods like block bootstrapping [6, 10, 17] or machine learning techniques. For instance, [3] employed an ensemble of GANs based on hierarchical clustering to sample high-dimensional financial time-series data, demonstrating that the synthetic data approximated the correlations found in the underlying empirical dataset.

Given their success in computer vision, DDPMs [14, 32] are promising candidates for sampling realistic correlation matrices. These models work by iteratively adding noise to input data and training a neural network to remove the noise and reconstruct the original data, allowing them to capture the underlying data distribution. DDPMs have achieved state-of-the-art results in various tasks, including image generation and text-to-image synthesis [4, 14, 29, 30].

2.2 Applications of Synthetic Correlation Matrices

Synthetic correlation matrices have found applications in various areas of quantitative finance. In asset allocation, Lopez De Prado introduced Hierarchical Risk Parity [22] and Nested Clustered Optimization [23], demonstrating their superior performance using synthetic correlation matrices. Following this, numerous studies [16, 26, 27] have utilized synthetic matrices to assess various asset allocation methods within simulation-based frameworks. For instance, [27] and [16] employed XGBoost models with SHAP framework [24] to identify key characteristics of correlation matrices that can influence selecting specific asset allocation methods.

[20] presented an approach to construct investment portfolios directly using synthetic correlation matrices. By providing a more diverse dataset while limiting outliers unlikely to occur in the future, this approach enhanced the robustness of the constructed portfolios to changing market conditions and demonstrated superior out-of-sample performance.

In addition to asset allocation, synthetic correlation matrices have been used for risk analysis. [1] proposed using generated correlation matrices to perform Monte Carlo simulations and estimate the Value-at-Risk surface over the latent space of a developed Variational Autoencoder.

Synthetic correlation matrices may also find applications in other areas, such as stress testing portfolios for unforeseen market conditions and pricing complex multi-asset derivatives [6].

2.3 Generating Supplementary Asset Attributes

For applications requiring information beyond correlation matrices, [20] proposed an Encoder-Decoder network that generates additional asset attributes (e.g., volatilities, expected returns) from correlation matrices. This approach recognizes that synthetic correlation matrices can provide insights into market structure and conditions. The network ensures consistency between input matrices and generated attributes, allowing for advanced modeling that would be limited if using only correlation data.

3 Methodology

3.1 Denoising Diffusion Probabilistic Models

Denoising diffusion probabilistic models [14, 32] learn to reverse a gradual corruption process that adds noise to data through multiple timesteps.

The forward diffusion process defines a Markov chain that incrementally adds Gaussian noise to the data x_0:

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \tag{1}$$

where t indexes the timesteps, $\beta_t \in (0,1)$ is a predefined noise schedule that controls the variance of the Gaussian noise added at each step, and $\mathcal{N}(x; \mu, \Sigma)$ denotes a normal distribution with mean μ and covariance matrix Σ. This process gradually transforms the data distribution, and for a sufficiently large number of steps T, the final distribution approaches a standard Gaussian distribution $x_T \approx \mathcal{N}(0, I)$.

The reverse process learns a function $p_\theta(x_{t-1}|x_t)$ that denoises x_t by estimating the previous timestep's data distribution:

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{2}$$

where μ_θ and Σ_θ are predicted by the model p_θ with parameters θ trained to denoise x_t. The training objective is to maximize the log-likelihood of the data under the generative model.

To generate samples, the model starts from pure noise $x_T \sim \mathcal{N}(0, I)$ and recursively denoises by sampling from $p_\theta(x_{t-1}|x_t)$ at each timestep, progressively reconstructing the data distribution of x_0.

For efficient training, following [14], the forward and reverse processes are reparameterized in terms of $\alpha_t = 1 - \beta_t$ and $\overline{\alpha}_t = \prod_{s=1}^{t} \alpha_s$:

$$q(x_t|x_0) := \mathcal{N}(x_t; \sqrt{\overline{\alpha}_t}x_0, (1-\overline{\alpha}_t)I) \tag{3}$$

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}\left(x_{t-1}; \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon_\theta(x_t, t)\right), \Sigma_\theta(x_t, t)\right) \tag{4}$$

where the model, instead of predicting μ_θ, learns the added noise $\epsilon_\theta(x_t, t)$ at timestep t. [14] found that $\Sigma_\theta(x_t, t)$ could be set to a constant $\sigma_t^2 I$, rather than being learned. Specifically, they proposed using either $\beta_t I$ or $\tilde{\beta}_t I$ for this constant, where $\tilde{\beta}_t = \frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\beta_t$. This modification reduces the model's complexity without significantly affecting its performance and results in a more efficient training loss for the reparameterized model:

$$L_{simple} = \mathbb{E}_{x_t,t}[||\epsilon_t - \epsilon_\theta(x_t, t)||_2^2] \tag{5}$$

where ϵ_t is the actual noise added at timestep t during the forward diffusion process, and $\epsilon_\theta(x_t, t)$ is the model's prediction of that noise.

3.2 Conditional Generation

Conditional denoising diffusion probabilistic models incorporate additional conditioning information y into the diffusion process to learn a conditional distribution $p_\theta(x_0|y)$ [4]. While the forward diffusion process mirrors the unconditional model, the reverse process includes y:

$$p_\theta(x_{t-1}|x_t, y) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t, y), \Sigma_\theta(x_t, t, y)) \tag{6}$$

To generate samples conditioned on y, we start from $x_T \sim \mathcal{N}(0, I)$ and recursively denoise through $p_\theta(x_{t-1}|x_t, y)$, recovering $p_\theta(x_0|y)$. The reparameterization and simplified loss from the unconditional model can be extended to the conditional case [30].

3.3 Implementation

Our implementation of the unconditional denoising diffusion model adheres to the methodology outlined by [14], originally devised for image applications. The model is based on U-net network [31] and Wide ResNet blocks of convolutional layers [35]. We applied group normalization [34] and attention layers [33] between the ResNet blocks, and applied a 0.5 dropout rate within the blocks. The model employs a linear beta schedule that increases from 0.0001 to 0.02, with a total of 1000 timesteps. Sinusoidal position embeddings [33] are used to encode the timesteps. We train the model for 2500 epochs using a step-based learning rate decay, starting with an initial learning rate of 0.001 and decreasing it by 30% every 100 epochs. The Adam optimizer [18] is used with $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

For conditional generation, two conditioning variables are transformed to sinusoidal position embeddings [33] and added within each upsample ResNet block.

Our code and further details on the model architectures for each dataset can be found in our GitHub repository[1].

3.4 Datasets

We use three empirical datasets to assess model performance:

(1) Futures: This dataset, based on the assets used in [27], comprises time-series data for 17 futures contracts representing commodity, fixed income, and equity indices. We compute 3-month correlation matrices using daily returns from May 2, 2000, to October 9, 2023, for each day in the dataset. The training dataset is selected to cover the period up to the end of 2015.
(2) Fixed income: Introduced in [20], this dataset consists of 68 assets, including 28 currencies and 40 investable fixed income indices, spanning from April 2007 to October 2022. The training data extends until March 2017. We calculate 1-year correlation matrices based on 52 weekly returns for each day in the dataset.
(3) Stocks: To evaluate performance on larger correlation matrices, we compiled a dataset of 183 MSCI Europe stocks that were included in the index as of October 18, 2023, and have been consistently listed for 30 years. We compute 2-year correlation matrices using daily returns from October 19, 1993, to October 18, 2023, for each day in the dataset. The training dataset extends until the end of 2013.

These datasets encompass diverse asset types and market structures, with matrix sizes ranging from 17x17 to 183x183. The training sections account for 62% to 65% of the entire datasets.

We employ two conditioning variables in the conditional diffusion model:

- Interest rates volatility (volatility of 10-year US treasury yield's differences)
- Equity volatility (volatility of S&P500 index's percentage returns)

For each dataset, periods for the conditioning volatility variables are aligned with the return periods of the corresponding correlation matrices.

All Bloomberg tickers to recreate the datasets are provided in our GitHub repository[1].

3.5 Model Evaluation

The evaluation process assesses the unconditional (DM) and conditional (CDM) denoising diffusion models' ability to generate realistic correlation matrices. We compare our models against empirical data and synthetic matrices produced by four alternative approaches:

(1) BB - block bootstrapping [6, 10, 17], a traditional finance approach. For the futures dataset, we use 3 blocks of 21 business days of returns; for fixed income, 4 blocks of 13 weekly returns; and for stocks, 8 blocks of 65 business days of returns.
(2) DCGAN - the original CorrGAN model based on [25].
(3) WGAN - a Wasserstein GAN model described in [20].
(4) VAE - a Variational Autoencoder that is inspired by [1].

All models were trained on the same training dataset and evaluated using both the training and testing datasets.

For each model, we generate a sample of 64,000 correlation matrices. To generate a dataset with the CDM model, we need to provide a distribution of conditioning variables. For each empirical dataset in our analysis, we generate two synthetic datasets using the model: CDM_Train, which is based on the distributions of conditioning variables from the empirical training datasets, and CDM_Test, based on the distributions from the testing datasets. The CDM_Train datasets can be compared with the empirical training datasets as well as synthetic datasets from the alternative machine learning models, whereas the CDM_Test datasets can be used as a

[1] https://github.com/szymkubiak/DDPM-for-Correlation-Matrices

more stringent evaluation to assess how well the model can replicate an unseen dataset.

The evaluation process unfolds in four stages:

(1) Examination of fundamental characteristics: diagonal elements forming a vector of ones and matrix symmetry. These are measured by absolute deviations of diagonal elements from 1 and absolute differences between upper and lower triangular parts, respectively.
(2) Evaluating Wasserstein distances between the generated and empirical datasets. This metric quantifies the dissimilarity between two probability distributions, with lower values indicating greater similarity. We calculate it by flattening each matrix into a vector and using the scipy.stats.wasserstein_distance function to compute the distance between the distributions of these flattened vectors.
(3) Assessing metrics suggested in [27] that align with key stylized facts observed in empirical correlation matrices:
   (a) Average correlation coefficient of matrices
   (b) Gini coefficient of matrices' eigenvalues
   (c) Cophenetic correlation between the original correlation distance matrix and the cophenetic matrix of the hierarchical clustering algorithm (using single or ward linkage). The distance is measured as $D = \sqrt{2(1 - Corr)}$
   (d) Sum of negative entries of the first eigenvector (related to the Perron-Frobenius property)
   (e) Power exponent of correlation matrices' eigenvalue distribution
(4) Additional evaluation of the conditional model: we partition each empirical training dataset into 10 subsets by splitting the distributions of the two conditioning variables into quintiles (5 equal-sized bins for each variable). Then, for each partition, we sample correlation matrices conditioned on the values of the corresponding conditioning variable within that partition's range. We assess the quality of the conditionally generated datasets by calculating Wasserstein distances between the sampled matrices and the empirical matrices within the same partition. Furthermore, we compare these partition-level distances to the distances between the CDM-generated partitions and the entire empirical training dataset, as well as the distances between the full DM-generated datasets and the partitions of the empirical training data.

Characteristics in point 1 are evaluated using the raw correlation matrices generated by all machine learning models. For subsequent assessments, we apply two simple transformations to ensure the matrices possess essential properties of correlation matrices. First, we enforce symmetry by using either the upper or lower triangular part of the matrix. Second, we set all diagonal elements to 1.

4 Experimental Results

4.1 Diagonal Elements and Matrix Symmetry

Table 1 presents the initial results of our analysis, focusing on the examination of diagonal elements and the symmetry of raw correlation matrices generated by the machine learning models. The BB method is excluded from this table as it directly samples from empirical data, ensuring perfect diagonal elements of 1 and symmetry by construction.

Table 1: Evaluation of diagonal elements and symmetry in raw correlation matrices generated by machine learning models (excluding BB method). "Diagonal Elements Abs Diffs from 1" measures how close diagonal elements are to 1; lower values indicate closer adherence to the ideal of unit diagonals. "Abs Diffs Between Triangular Matrices" quantifies the differences between upper and lower triangular parts; lower values indicate higher symmetry. The WGAN model consistently achieves the lowest absolute deviations from 1 for diagonal elements across all datasets. The DM model produces the most symmetric matrices.

	Diagonal Elements Abs Diffs from 1		Abs Diffs Between Triangular Matrices	
	Avg	Std	Avg	Std
Futures Dataset				
DCGAN	0.0176	0.0107	0.0377	0.0593
WGAN	**0.0010**	0.0011	0.0441	0.0626
VAE	0.0013	**0.0010**	**0.0054**	**0.0072**
DM	0.0023	0.0018	0.0061	0.0085
CDM_Train	0.0033	0.0026	0.0095	0.0126
CDM_Test	0.0033	0.0027	0.0096	0.0127
Fixed Income Dataset				
DCGAN	0.0281	0.0179	0.1316	0.1872
WGAN	**0.0008**	**0.0007**	0.0663	0.0930
VAE	0.0021	0.0024	0.0404	0.0530
DM	0.0039	0.0033	**0.0158**	**0.0215**
CDM_Train	0.0095	0.0079	0.0295	0.0390
CDM_Test	0.0095	0.0079	0.0295	0.0390
Stocks Dataset				
DCGAN	0.0377	0.0239	0.1260	0.1699
WGAN	**0.0012**	**0.0010**	0.0644	0.0857
VAE	0.0022	0.0046	0.0212	0.0273
DM	0.0063	0.0051	**0.0089**	**0.0115**
CDM_Train	0.0134	0.0107	0.0189	0.0243
CDM_Test	0.0142	0.0111	0.0202	0.0256

Across all three datasets, the WGAN model consistently achieves the lowest average absolute deviation of diagonal elements from 1, as well as the lowest standard deviation of these deviations. The VAE model follows closely, while the DM model's deviations are larger than WGAN and VAE but smaller than DCGAN. Notably, the DM model's diagonal elements can deviate both above and below one, while other models never exceed this value due to their output being capped by the Tanh activation function.

When assessing matrix symmetry, DM significantly outperforms in the fixed income and stocks markets, which feature larger matrices. It provides the lowest absolute differences between values of upper and lower triangular matrices and also exhibits the lowest standard deviations of these differences. However, in the futures dataset, the VAE model moderately outperforms the DM model.

Table 2: Wasserstein distances between generated and empirical datasets for futures (FUT), fixed income (FI), and stocks (EQ). The DM model achieves the lowest distances to the empirical training datasets across all markets. For empirical testing datasets, CDM_Test achieves the lowest distance for stocks but not for futures and fixed income.

	Distance to Empirical Training			Distance to Empirical Testing		
	FUT	FI	EQ	FUT	FI	EQ
Empirical Training	-	-	-	0.030	0.053	0.078
Empirical Testing	0.030	0.053	0.078	-	-	-
BB	0.006	0.017	0.035	0.029	0.069	0.082
DCGAN	0.004	0.012	0.007	0.027	0.064	0.084
WGAN	0.005	0.015	0.006	0.027	0.053	0.079
VAE	0.021	0.021	0.024	**0.023**	0.064	0.080
DM	**0.001**	**0.003**	**0.002**	0.029	**0.050**	0.080
CDM_Train	0.011	0.005	0.031	0.038	0.058	0.109
CDM_Test	0.011	0.006	0.035	0.027	0.059	**0.055**

The CDM model, according to these two metrics, provides lower-quality matrices than DM, but it outperforms DCGAN and WGAN in terms of matrix symmetry.

4.2 Wasserstein Distances

As shown in Table 2, the DM model's generated datasets exhibit significantly lower Wasserstein distances to the empirical training datasets compared to alternative generative models. Although the CDM_Train model shows a deterioration in performance relative to DM, it remains competitive for the futures and fixed income datasets, despite performing worse on the stocks dataset.

Notably, CDM_Test provides the lowest distance to the empirical testing dataset for stocks. However, for the fixed income market, CDM_Test exhibits higher distance to their respective empirical testing dataset compared to the training dataset. This behavior suggests that either overfitting to the training data distributions is occurring, or that additional conditioning variables beyond the two selected ones may be required to more closely replicate the distributions of the empirical testing datasets across all markets.

4.3 Stylized Facts of Correlation Matrices

Table 3 provides a comprehensive analysis of stylized facts for the generated correlation matrices across models. Ideally, DM and CDM_Train would embody the key characteristics present in the training datasets, whereas CDM_Test would resemble the characteristics of the empirical testing datasets.

The DM model consistently demonstrates superior performance across all three markets, with its advantages particularly pronounced for larger correlation matrices. It generates matrices with correlation coefficients closely mirroring the training empirical distributions, and excels in producing matrices with the most realistic eigenvalue and eigenvector distributions. Analysis of the cophenetic correlations highlights DM's ability to capture the hierarchical correlation structure remarkably similar to the training data. While the CDM_Train offers competitive matrices, its metrics are moderately worse than DM's but often better than the other generative models.

For stocks, CDM_Test provides matrices whose mean metrics are closer to the testing data than CDM_Train. When a metric's average is higher in the empirical testing data versus training, this is mostly reflected in CDM_Test having a higher value for that metric compared to CDM_Train. However, for futures and fixed income, CDM_Test does not exhibit this behavior, with its metrics more closely resembling the training data. This also suggests potential overfitting or that the two conditioning variables alone may be insufficient to fully capture the distribution shifts of these correlation matrices over time.

Despite using empirical data directly to construct samples, the BB method provided worse metrics than most alternative methods, demonstrating the superior performance of machine learning models in replicating the stylized facts of empirical datasets.

4.4 Evaluation of the Conditional Generation

Table 4 presents Wasserstein distances evaluating how well the CDM model captures distributions of correlation matrices across different volatility regimes. Across all three datasets, CDM_Train matrices conditioned on specific volatility quintiles exhibit significantly lower average distances to the corresponding empirical partitions compared to the full empirical training datasets. The distances were also much lower than of the entire DM model's dataset to the empirical training partitions. This indicates the CDM model to effectively separate the correlation matrices' distributions based on volatility regimes.

However, the model's performance varies across quintiles. For the highest volatility quintile (5th), distances between CDM_Train partitions and empirical partitions are moderately lower than distances for other quintiles, demonstrating strong ability to capture correlation dynamics during high volatility periods. Conversely, for other volatility quintiles (1st-4th), the distance improvements are sometimes less pronounced, with some cases not outperforming the unconditional model's (DM) distances.

4.5 Discussion and Implications of Results

Our empirical analyses demonstrate the superior performance of DDPMs in generating realistic financial correlation matrices across diverse markets. The DM model consistently outperformed alternative approaches, exhibiting the lowest Wasserstein distances to empirical training datasets and effectively reproducing key stylized facts of empirical correlation matrices, particularly for larger matrices, suggesting scalability.

The conditional diffusion model (CDM) showed promising results in generating correlation matrices reflecting specific market volatility regimes, with notably strong performance in high volatility conditions. This suggests particular efficacy in capturing correlation dynamics during market stress periods.

An important observation from our empirical work, not previously mentioned, is the stability of the training process for both DM and CDM models. This stands in contrast to our experience with WGAN and DCGAN models, where we encountered the standard instability issues commonly associated with GAN training.

Table 3: Summary statistics of stylized facts for the empirical and generated correlation matrices across the three datasets. The table presents the mean and standard deviation of six metrics: mean correlation coefficient (mean correl), Gini coefficient of eigenvalues (eigen gini), cophenetic correlation with single and Ward's linkage (coph corr single/ward), sum of negative entries in the first eigenvector (perron frob sum neg), and power law exponent of eigenvalue distribution (power eigen values). Highlighted values indicate the closest match to the empirical training datasets for each metric. The DM model demonstrates the best overall performance, producing correlation matrices with stylized facts most similar to the empirical data.

	mean correl		eigen gini		coph corr single		coph corr ward		perron frob sum neg		power eigen values	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Futures Dataset												
Empirical Training	0.152	0.046	0.606	0.057	0.899	0.045	0.859	0.054	1.530	0.508	2.142	0.443
Empirical Testing	0.171	0.051	0.591	0.041	0.877	0.053	0.837	0.049	1.577	0.513	2.256	0.441
BB	0.149	0.039	0.607	0.045	0.897	0.038	0.863	0.042	1.539	0.520	2.109	0.412
DCGAN	0.151	0.045	0.606	0.059	0.893	0.047	0.856	0.055	1.609	0.520	2.223	0.395
WGAN	0.158	0.042	0.621	0.065	0.897	0.048	**0.858**	0.054	1.624	0.503	2.181	0.394
VAE	0.149	0.038	0.561	0.063	0.944	0.019	0.894	0.035	1.237	0.532	2.219	0.289
DM	0.153	**0.046**	**0.606**	**0.057**	0.901	**0.045**	0.861	**0.054**	**1.524**	**0.509**	**2.134**	0.431
CDM_Train	0.149	0.048	0.619	0.060	**0.898**	0.049	0.861	0.056	1.564	0.524	2.131	**0.442**
CDM_Test	**0.152**	0.042	0.602	0.065	0.896	0.044	0.857	0.051	1.609	0.517	2.204	0.567
Fixed Income Dataset												
Empirical Training	0.231	0.063	0.855	0.011	0.716	0.118	0.751	0.071	2.335	0.389	1.952	0.093
Empirical Testing	0.284	0.079	0.862	0.019	0.717	0.102	0.697	0.079	2.227	0.396	1.965	0.112
BB	0.216	0.051	**0.851**	0.012	0.704	0.110	0.752	0.056	**2.414**	**0.404**	1.955	0.100
DCGAN	0.222	0.038	1.044	0.031	0.670	0.098	0.728	0.077	3.306	0.288	2.154	0.068
WGAN	**0.230**	0.070	0.976	0.064	0.697	0.115	0.742	0.073	3.225	0.303	2.032	0.081
VAE	0.220	0.049	0.865	0.019	0.730	0.098	0.767	0.059	3.246	0.278	2.065	0.076
DM	0.234	**0.064**	0.864	**0.012**	**0.716**	**0.120**	0.750	0.070	3.191	0.293	**1.952**	**0.093**
CDM_Train	0.226	0.062	0.880	0.014	0.712	0.122	**0.751**	**0.071**	3.332	0.296	1.967	0.089
CDM_Test	0.225	0.062	0.880	0.014	0.705	0.125	0.751	0.073	3.322	0.294	1.965	0.090
Stocks Dataset												
Empirical Training	0.276	0.116	0.619	0.086	0.833	0.061	0.302	0.136	5.128	0.348	2.995	0.272
Empirical Testing	0.355	0.062	0.716	0.048	0.753	0.030	0.377	0.097	5.094	0.318	2.498	0.163
BB	0.273	0.062	0.631	0.046	0.803	0.045	0.274	0.103	5.201	0.314	2.904	0.163
DCGAN	0.273	0.076	0.850	0.092	0.661	0.103	0.290	0.122	5.336	0.300	3.186	0.203
WGAN	0.278	0.117	0.769	0.151	0.763	0.064	0.268	0.108	5.253	0.308	3.214	0.312
VAE	**0.275**	0.090	0.600	0.083	0.840	0.036	0.264	0.111	**5.132**	0.329	3.168	0.223
DM	0.274	**0.115**	**0.617**	**0.086**	**0.836**	**0.062**	**0.305**	0.131	5.152	**0.347**	**3.017**	0.265
CDM_Train	0.245	0.112	0.606	0.088	0.822	0.074	0.294	**0.137**	5.217	0.337	3.144	**0.279**
CDM_Test	0.299	0.077	0.639	0.056	0.844	0.037	0.315	0.098	5.196	0.308	3.097	0.164

The ability to generate high-quality synthetic correlation matrices has significant implications for risk management and asset allocation. Our DDPM-based approach offers a method to augment historical data with realistic synthetic samples, potentially leading to more robust risk assessments and portfolio strategies. This may be particularly valuable for stress testing and scenario analysis. The quality of the generated data suggests that DDPMs could help address the scarcity of financial data, especially for extreme events, potentially accelerating the development and testing of new financial models and strategies.

5 Limitations

While our approach demonstrates strong empirical performance, it faces several challenges. The model's scalability is constrained by computational requirements, especially for larger correlation matrices that require substantial GPU resources. Adding new assets or extending the time series necessitates retraining the entire model, which can be time-consuming. An additional limitation is the generation speed of DDPMs, which is generally slower compared to some other machine learning models. This slower speed may restrict their applicability in scenarios requiring real-time or high-frequency data generation.

Despite the promising empirical results, the theoretical justification for DDPMs' effectiveness in generating correlation matrices remains limited. Further research is needed to understand why these models perform well in this context. Moreover, correlation matrices inherently reduce complex asset relationships to single values, potentially oversimplifying non-linear interactions that might be better captured by more sophisticated models.

Table 4: Wasserstein distances evaluating the CDM model's ability to capture volatility regime dynamics. Distances are reported between: 1) CDM_Train and empirical matrices within each volatility quintile partition, 2) full DM-generated datasets and empirical matrices within volatility quintiles, and 3) CDM_Train matrices partitioned by volatility quintiles and the entire empirical training dataset. Lower distances, which are highlighted, indicate better alignment of the distributions. The results show that the CDM_Train partitions have much lower average distances to the corresponding empirical training partitions, suggesting the CDM model effectively captures empirical data characteristics across different market volatility conditions.

	Rates Volatility Quintile						Equity Volatility Quintile					
	1	2	3	4	5	Mean	1	2	3	4	5	Mean
Futures Dataset												
CDM_Train Partition vs Emp. Tr. Partition	**0.013**	0.009	0.015	**0.011**	**0.013**	**0.012**	**0.006**	**0.004**	0.014	**0.020**	**0.017**	**0.012**
All DM Data vs Emp. Tr. Partition	0.031	0.011	**0.005**	0.016	0.043	0.021	0.054	0.027	**0.010**	0.029	0.060	0.036
CDM_Train Partition vs All Emp. Tr. Data	0.029	**0.007**	0.013	0.023	0.052	0.025	0.055	0.024	0.011	0.042	0.073	0.041
Fixed Income Dataset												
CDM_Train Partition vs Emp. Tr. Partition	**0.009**	**0.008**	**0.003**	**0.003**	**0.004**	**0.006**	**0.008**	**0.009**	**0.002**	**0.005**	**0.004**	**0.006**
All DM Data vs Emp. Tr. Partition	0.027	0.040	0.012	0.032	0.051	0.032	0.086	0.015	0.021	0.034	0.050	0.041
CDM_Train Partition vs All Emp. Tr. Data	0.021	0.034	0.012	0.032	0.051	0.030	0.080	0.009	0.020	0.036	0.050	0.039
Stocks Dataset												
CDM_Train Partition vs Emp. Tr. Partition	0.038	**0.044**	**0.025**	0.036	**0.010**	**0.031**	**0.019**	0.046	0.055	**0.024**	**0.009**	**0.031**
All DM Data vs Emp. Tr. Partition	**0.031**	0.053	0.093	0.055	0.086	0.064	0.067	**0.012**	0.056	0.051	0.078	0.053
CDM_Train Partition vs All Emp. Tr. Data	0.032	0.099	0.121	**0.017**	0.074	0.069	0.089	0.052	**0.008**	0.077	0.066	0.058

While our models replicate historical correlation structures effectively, their ability to generalize to future market relationships remains uncertain. Financial markets are dynamic and prone to structural changes or unforeseen events, which may limit the models' capacity to adapt, especially during significant market shifts. Furthermore, the conditional diffusion model (CDM) uses only two conditioning variables — rates and equity volatility — potentially overlooking other important factors that influence correlation structures across various market regimes.

Finally, the black-box nature of DDPMs poses challenges to interpretability compared to traditional financial methods. This lack of transparency could be problematic in regulated financial environments where model explainability is crucial.

6 Practical Application

Consider an investor evaluating an asset allocation strategy within the fixed-income universe. The strategy aims to minimize portfolio volatility while targeting a minimum expected return of 1% above the 3-month U.S. Treasury bill rate. The investor seeks to understand the expected volatility level for this strategy and how it might vary across different market regimes. To address this, the analysis compares the insights gained from using a synthetic dataset generated by the CDM model against relying solely on historical empirical data.

Following [20], we extended our fixed income dataset beyond correlation matrices to include additional attributes. The gathered empirical dataset covers the same period and assets as the fixed income training dataset from previous sections but consists of 1-year correlation matrices, 1-year volatilities, expected returns over the next 1-year (including annualized 3-month Treasury bill rate), and 4-week forward realized returns. The corresponding synthetic dataset is generated as follows:

- Correlation matrices are sampled from the CDM model, with the conditioning variables (rates and equity volatility) distributed uniformly across their ranges in the empirical dataset. In cases where a generated correlation matrix was not positive definite, we applied the adjustment method outlined by [13].
- The remaining components (volatilities, expected returns, forward returns) are sampled from an Encoder-Decoder model [20] (described in the "Related Work" section) that generates these asset attributes conditional on a given correlation matrix.

For each correlation matrix C and vector of volatilities v, a covariance matrix Σ is calculated as $\Sigma = \text{diag}(v) \cdot C \cdot \text{diag}(v)$. Then, for each covariance matrix Σ and vector of expected returns μ, portfolio weights w are optimized to minimize the portfolio's variance ($w^T \Sigma w$) subject to constraints: portfolio expected return $\geq$ U.S. 3-month Treasury bill rate + 1%, sum of bond indices weights = 1, weight of each bond index ≥ 0, and weight of each currency between -5% and 5%.

Subsequently, the realized returns of the optimized portfolios are observed over the following 4 weeks and grouped into equity and volatility buckets for analysis of the strategy's performance across different market regimes.

Figure 1 illustrates the observation count and portfolios' annualized realized volatility across the rates and equity volatility buckets. The empirical dataset's heatmaps reveal that most observations correspond to periods when both volatilities were low, resulting in a sparse representation of the strategy's performance across different market conditions. The synthetic dataset offers two main benefits

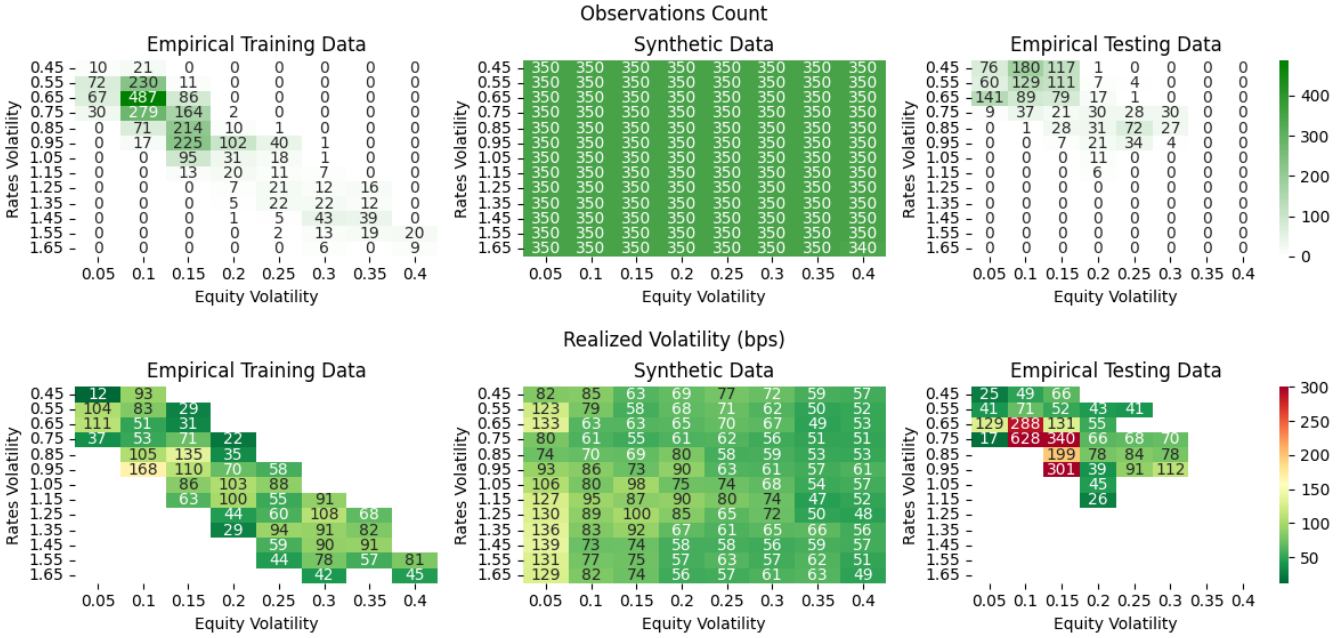

Figure 1: Heatmaps illustrating observations count and annualized realized volatility of the investor's strategy across different rates and equity volatility regimes in synthetic, empirical training, and empirical testing datasets. The ticks on the x and y axes correspond to lower boundaries of conditioning variables' buckets. The synthetic dataset provides a more smooth and monotonic "surface". Additionally, it provides insights for regimes not covered by the empirical dataset.

Table 5: Performance statistics of XGBoost models in the testing period to predict realized volatility of the strategy. The models are trained with either the empirical training or the synthetic dataset. MAE stands for mean absolute error, and RMSE for root mean square error. The 10th, 25th, 50th (median), 75th, and 90th percentile columns show the distribution of absolute errors for each dataset variant. The Diebold-Mariano (DM) test statistic and its corresponding p-value are reported for comparing forecast accuracy, with a negative DM statistic indicating superior performance of the Synthetic variant. The DM test uses mean squared error as the criterion and evaluates predictions made 20 steps ahead, corresponding to a 4-week forecasting horizon. All error metrics are reported in basis points of annualized volatility.

	MAE	RMSE	10th Perc. Abs Error	25th Perc. Abs Error	Median Abs Error	75th Perc. Abs Error	90th Perc. Abs Error	DM Stat	p-value
Empirical	63	158	7	16	35	61	111	–	–
Synthetic	57	152	7	19	37	57	78	-3.19	0.1%

over the empirical dataset. Firstly, it presents a smoother and more monotonic "surface" of the strategy's performance, making it easier to identify trends and patterns across different market regimes. Secondly, the synthetic dataset provides insights into the strategy's behavior in market conditions that are not well-represented or absent in the empirical dataset, allowing for a more comprehensive analysis of the strategy's performance.

To further assess the value of the synthetic dataset in predicting the strategy's realized volatility, an off-the-shelf XGBoost model [2] is employed. Two separate XGBoost models are trained - one on the empirical training dataset, and another on the synthetic dataset. Both models use the two conditioning variables (rates and equity volatility) as input features to predict the strategy's annualized volatility of returns over a 4-week horizon.

Table 5 compares the performance of these XGBoost models on the testing period data. The model trained on the synthetic dataset demonstrates lower mean absolute error (57 vs 63 basis points). The improved performance of the synthetic variant is further evidenced by a Diebold-Mariano test statistic [5, 12] of -3.19 (p-value: 0.1%) using mean squared error as the criterion for 20-step ahead predictions, corresponding to a 4-week volatility forecasting horizon. These results suggest that the synthetic dataset, by providing a more comprehensive representation of market conditions, enables more accurate predictions of the investment strategy's realized volatility across different regimes.

The practical application section demonstrates that the synthetic dataset generated by the CDM model can help mitigate the issue of limited financial data and provide better estimates of the strategy's

performance. By leveraging synthetic data, an investor can gain a more comprehensive understanding of the strategy's behavior across a wider range of market conditions and make more informed decisions regarding portfolio construction and risk management.

7 Conclusion

This paper explores the application of denoising diffusion probabilistic models (DDPMs) to generate realistic synthetic correlation matrices for financial modeling, with a focus on futures, fixed income, and equity markets. We propose an unconditional diffusion model (DM) and a conditional diffusion model (CDM) that incorporates market conditions through volatility-based conditioning variables. Our experimental results demonstrate that the DM model consistently outperforms alternative methods in generating correlation matrices that closely resemble the empirical training datasets, while the CDM model effectively captures the differences in correlation matrix distributions across various volatility regimes.

A case study focusing on an asset allocation strategy showcases the practical utility of our approach. By using synthetic data generated by the CDM model, we obtain a more comprehensive representation of the strategy's performance across a broader range of market conditions than when using only empirical data.

Our findings highlight DDPMs' potential to overcome the limitations of scarce financial data. By augmenting empirical datasets with plausible synthetic data, we can enhance the analysis of financial strategies, improve risk assessment, and support more informed decision-making in quantitative finance. Future research could explore additional conditioning variables and extend DDPM applications to other financial domains, such as stress testing and scenario analysis, for further insights and benefits.

References

[1] Sergio Caprioli, Emanuele Cagliero, and Riccardo Crupi. 2023. Quantifying Credit Portfolio sensitivity to asset correlations with interpretable generative neural networks. *arXiv preprint arXiv:2309.08652* (2023).

[2] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 785–794.

[3] Fernando de Meer Pardo, Peter Schwendner, and Marcus Wunsch. 2022. Tackling the Exponential Scaling of Signature-Based Generative Adversarial Networks for High-Dimensional Financial Time-Series Generation. *The Journal of Financial Data Science* 4, 4 (2022), 110–132.

[4] Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems* 34 (2021), 8780–8794.

[5] Francis X Diebold and Robert S Mariano. 2002. Comparing predictive accuracy. *Journal of Business & economic statistics* 20, 1 (2002), 134–144.

[6] Matthias R Fengler and Peter Schwendner. 2004. Quoting multiasset equity options in the presence of errors from estimating correlations. *Journal of Derivatives* 11, 4 (2004), 43.

[7] Soumyadip Ghosh and Shane G Henderson. 2003. Behavior of the NORTA method for correlated random vector generation as the dimension increases. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 13, 3 (2003), 276–294.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

[9] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. *Advances in neural information processing systems* 30 (2017).

[10] Peter Hall. 1985. Resampling a coverage pattern. *Stochastic processes and their applications* 20, 2 (1985), 231–246.

[11] Johanna Hardin, Stephan Ramon Garcia, and David Golan. 2013. A method for generating realistic correlation matrices. *The Annals of Applied Statistics* (2013), 1733–1762.

[12] David Harvey, Stephen Leybourne, and Paul Newbold. 1997. Testing the equality of prediction mean squared errors. *International Journal of forecasting* 13, 2 (1997), 281–291.

[13] Nicholas J Higham. 1988. Computing a nearest symmetric positive semidefinite matrix. *Linear algebra and its applications* 103 (1988), 103–118.

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

[15] Amelie Hüttner, Jan-Frederik Mai, and Stefano Mineo. 2018. Portfolio selection based on graphs: Does it align with Markowitz-optimal portfolios? *Dependence Modeling* 6, 1 (May 2018), 63–87. https://doi.org/10.1515/demo-2018-0004

[16] Markus Jaeger, Stephan Krügel, Dimitri Marinelli, Jochen Papenbrock, and Peter Schwendner. 2020. Understanding Machine Learning for Diversified Portfolio Construction by Explainable AI. *SSRN Electronic Journal* (2020). https://doi.org/10.2139/ssrn.3528616

[17] Markus Jaeger, Stephan Krügel, Dimitri Marinelli, Jochen Papenbrock, and Peter Schwendner. 2021. Interpretable machine learning for diversified portfolio construction. *The Journal of Financial Data Science* 3, 3 (2021), 31–51.

[18] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[19] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[20] Szymon Kubiak, Tillman Weyde, Oleksandr Galkin, Dan Philps, and Ram Gopal. 2023. Improved Data Generation for Enhanced Asset Allocation: A Synthetic Dataset Approach for the Fixed Income Universe. *arXiv preprint arXiv:2311.16004* (2023).

[21] Daniel Lewandowski, Dorota Kurowicka, and Harry Joe. 2009. Generating random correlation matrices based on vines and extended onion method. *Journal of multivariate analysis* 100, 9 (2009), 1989–2001.

[22] Marcos Lopez de Prado. 2016. Building diversified portfolios that outperform out-of-sample. *Journal of Portfolio Management* (2016).

[23] Marcos Lopez de Prado. 2016. A robust estimator of the efficient frontier. *Available at SSRN 3469961* (2016).

[24] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30 (2017).

[25] Gautier Marti. 2020. Corrgan: Sampling realistic financial correlation matrices using generative adversarial networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 8459–8463.

[26] Gautier Marti, Victor Goubet, and Frank Nielsen. 2021. cCorrGAN: Conditional correlation GAN for learning empirical conditional distributions in the elliptope. In *Geometric Science of Information: 5th International Conference, GSI 2021, Paris, France, July 21–23, 2021, Proceedings 5*. Springer, 613–620.

[27] Jochen Papenbrock, Peter Schwendner, Markus Jaeger, and Stephan Krügel. 2021. Matrix evolutions: synthetic correlations and explainable machine learning for constructing robust investment portfolios. *Jochen Papenbrock, Peter Schwendner, Markus Jaeger and Stephan Krügel The Journal of Financial Data Science Spring* (2021).

[28] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015).

[29] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. [n. d.]. Hierarchical text-conditional image generation with clip latents. ([n. d.]).

[30] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 234–241.

[32] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*. PMLR, 2256–2265.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[34] Yuxin Wu and Kaiming He. 2018. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*. 3–19.

[35] Sergey Zagoruyko and Nikos Komodakis. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146* (2016).