

Universiteit Leiden

Master Computer Science

Towards Understanding Semantic Artifacts in Training Data for AI-Generated Image Detection

Name:	Steijn Lek
Student ID:	4029453
Date:	30-4-2026
Specialisation:	Artificial Intelligence
1st supervisor:	Dr. H.R. Doughty
2nd supervisor:	Dr. D. Soydaner Ph.D.

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Contents

1 Introduction **5**

2 Related work **8**
2.1 Image generation models . . . 8
2.2 Detecting synthetic images . . . 8
2.3 Bias in training datasets . . . 9
2.4 Semantic Artifacts . . . 10

3 Methodology **11**
3.1 Overall method . . . 11
3.2 Establishing the Semantic Artifact (SEMAR) score . . . 11
3.2.1 Noise residual power spectra and Artifacts . . . 11
3.2.2 Angular variance . . . 15
3.2.3 Pipeline - Creating the power spectrum . . . 16
3.2.4 Pipeline - Calculating angular variance . . . 17
3.2.5 Pipeline - Calculating the SEMAR score . . . 17
3.3 Detecting generated images . . . 19
3.3.1 Data generation . . . 19
3.3.2 Preprocessing . . . 19
3.3.3 Training the classifiers . . . 20

4 Experiments and results **21**
4.1 Datasets . . . 21
4.1.1 Dataset selection and construction . . . 21
4.1.2 Datasets . . . 21
4.1.3 Dataset generation details . . . 22
4.2 Implementation details . . . 22
4.2.1 Training details . . . 22
4.2.2 Evaluation . . . 22
4.3 A quantitative measurement for semantic artifacts . . . 23
4.3.1 Results . . . 23
4.3.2 Discussion . . . 24
4.4 Semantic artifacts and model generalization . . . 26
4.4.1 Results . . . 26
4.4.2 Discussion . . . 27
4.5 Patch shuffling and model generalization . . . 28
4.5.1 Results . . . 28
4.5.2 Discussion . . . 29
4.6 JPEG compression and model generalization . . . 31
4.6.1 Results . . . 31
4.6.2 Discussion . . . 32

5 Limitations **33**
5.1 Reliability of the SEMAR score . . . 33
5.1.1 The influence of data preprocessing on Semantic Artifacts . . . 33

6 Conclusion **35**

A Appendix **39**
A.1 Dataset details 39
A.2 Dataset generation details 39

Abstract

With the rise in popularity and sophistication of AI-generated images, the ability to detect and distinguish these images from real ones is becoming a more important challenge. Although current classifier models are able to accurately detect images that fall within the same distribution as the training data, generalization remains a significant challenge. Semantic artifacts refer to a specific type of bias in an image dataset, originating from the semantics of the images such as the type of object, scene or composition. During training, classifier models could overfit on this type of artifact, potentially hurting generalization. This study further investigates semantic artifacts and their impact on model generalization. A method is established by which semantic artifacts in a dataset can be described in a quantitative way. The method measures the change in distribution of energy in a specific area of the frequency power spectrum of a dataset when applying patch shuffling to the dataset. This quantification is used to investigate a correlation between the level of semantic artifacts in a dataset and the generalization of a model when trained on this dataset. Findings suggest that semantic artifacts in a dataset can be described in a quantitative way. Moreover, a negative correlation is visible between this type of artifact and model generalization. Patch shuffling of the input images is shown to reduce overfitting of classification models on this type of artifact. These outcomes could contribute to the field of AI-generated image detection by providing more insight into the role of semantic artifacts in model generalization, ultimately supporting the development of more robust detection models.

1 Introduction

Artificial Intelligence (AI) is playing a large role in modern society. It has seen a rapid evolution in recent years, and is being adopted more and more by companies, individuals and governments [1, 2]. AI offers numerous benefits, such as improving innovation, efficiency and productivity in both work and private life [3]. In recent years, there has been a significant rise in the use of AI for generating artificial images. This application of AI poses several risks. It enables the creation and spread of misinformation, for example by creating footage of events that never occurred. This could have serious consequences, such as threatening political stability and overall trust in digital content [4]. Moreover, this technology provides the foundation for the creation of deepfakes, in which existing media, such as videos or images, are manipulated in a potentially harmful way using deep learning-based approaches [5]. Common cases of deepfakes include the manipulation of digital content of celebrities or politicians, for example by altering their speech, actions or gestures [6]. The impact of these risks highlights the importance of being able to detect images that are created or altered using AI.

Much research has already been conducted on this topic, mainly focusing on detecting specific cues and signals from the generation process of AI-generated content [7, 8, 9]. These studies have shown that recent image generation models still leave behind unique and traceable artifacts in their outputs, which can be exploited to train classifiers for detecting AI-generated images. These artifacts are commonly referred to as generator artifacts. Nevertheless, current classifiers still suffer from large drops in performance when tested on different datasets and new types of generative models [10]. This drop in generalization is often a consequence of the classifiers overfitting on the generator artifacts of a specific generative model [11, 8].

While generalization remains a major bottleneck in the field of AI-generated image detection, several studies have demonstrated improvements in this area. These approaches remain relevant for state-of-the-art generative models like Latent Diffusion Models (LDMs) [12]. For example, Ojha et al. [13] show an improvement in generalization by mapping images into the feature space of a large vision-language model. Using techniques such as nearest-neighbor classification and linear probing, images can be classified as either real or AI-generated. Other studies focus on improving the quality of training data. They do this by reducing bias through controlling the image generation process [14] or adding annotations and explanations to the samples in the dataset [15].

A more recent topic in the generalization of AI-generated image detection models is the concept of semantic artifacts, introduced in a study by Zheng et al. [16]. The authors identify a key problem in AI-generated image detection, which is that the performance of classifier models tends to drop significantly when tested on datasets with different scenes and semantic content. Moreover, the authors show that, even if datasets are generated by the same generative model, models tend to suffer from poor generalization when evaluated across different image scenes. They attribute this poor generalization to models overfitting on semantic artifacts: a specific type of artifact pattern that is visible in the frequency spectrum of the dataset, and is caused by the semantic aspects of its images. These aspects could include the general structure, composition, color and type of object present in the images of the dataset. The artifacts arising from these semantic aspects are present in both real and generated images, as generative models tend to inherit them from the data they are trained on. This can cause classifier models to overfit on these scene-specific artifact patterns instead of learning features that generalize across scenes. For example, a model trained on a dataset consisting of images of bedrooms generated by an LDM could fail to detecti images of churches generated by the same model [16]. This shows that the model learns the scene-specific cues to distinguish a real image

from a generated one, rather than learning universal cues to detect generated images across different scenes.

Although Zheng et al. demonstrate ways to mitigate the overfitting on semantic artifacts, the concept of semantic artifacts remains relatively unexplored. First, no method yet exists to quantitatively describe semantic artifacts in a dataset, a limitation which is also mentioned in the appendix of the study by Zheng et al. Moreover, the impact of semantic artifacts on model generalization, especially compared to other artifacts coming from the generative models or preprocessing methods, still remains a topic to be explored more in-depth.

The goal of this study is to address these gaps by combining an analysis of the results of prior work on semantic artifacts with additional experiments. First, we establish a novel method for describing semantic artifacts in a dataset. This method analyzes how the distribution of energy in a specific area of the frequency power spectrum of a dataset changes when applying patch shuffling to the images. By applying this method to a dataset, its level of semantic artifacts can be captured in a single metric. We propose this novel metric as the semantic artifact score, hereafter referred to as the **SEMAR** score. The SEMAR score is computed for a set of five different datasets. A separate classification model is trained on each of these datasets, after which the generalization of the model is analyzed by examining its performance on the other datasets. This way, insight into the relationship between the SEMAR score and generalization of a dataset can be established. Finally, in extension of the study by Zheng et al. [16], the effects of patch shuffling on the SEMAR score and model generalization are investigated. This aims to validate whether patch shuffling could mitigate the overfitting of classifier models on semantic artifacts in training data.

As mentioned earlier, the study by Zheng et al. also includes artifacts from preprocessing methods in its definition of semantic artifacts. This study, however, defines semantic artifacts as artifacts solely related to the semantics of the images and not to the preprocessing methods. A clear distinction between these two types of artifacts will be maintained during the experiments in this study. This approach aims to provide a clearer understanding of semantic artifacts and how they differ from artifacts originating from other sources. Moreover, to ensure consistency and comparability with recent developments in the field of AI-generated image detection, Latent Diffusion Models are used to generate the synthetic data in all experiments.

This study aims to answer the following three research questions:

1. How can semantic artifacts in a dataset be described in a quantitative way?
2. How do semantic artifacts in the training data affect a model's ability to generalize?
3. What is the impact of applying patch shuffling to a dataset on its level of semantic artifacts and the generalization of a model trained on it?

Based on the research questions above, we hypothesize that semantic artifacts can be observed and measured most clearly in the frequency power spectrum of a dataset. Furthermore, we expect that the level of semantic artifacts in training data is negatively correlated with the generalization of a model trained on that dataset. In other words, the more semantic artifacts exist in a dataset, the less a model trained on that dataset will generalize well to other datasets. This drop in generalization is expected to be most significant when there is a large difference in semantics between the images from the training dataset and the test datasets. Finally, we expect that applying patch shuffling to the training images improves model generalization compared to not using patch shuffling. It is expected that this positive impact becomes more apparent as the level of semantic artifacts in the training dataset increases.

Our work provides three main contributions:

1. Improved understanding of semantic artifacts in datasets by developing and applying a novel quantitative metric. The motivation for this stems directly from one of the main points described in the limitations of [16]. This would give a better insight into what semantic artifacts actually are, what factors influence them and how datasets with different levels of semantic artifacts look like.

2. A novel investigation into whether a causal relationship exists between the level of semantic artifacts in a dataset and the generalization of a model trained on that dataset. This would show the direct effects of semantic artifacts in datasets for training classifier models.

3. An additional investigation of the impact of patch shuffling on the level of semantic artifacts in a dataset, as well as on model performance and generalization.

2 Related work

The following section aims to establish a clear outline of what relevant research has been done on the topics of AI-generated image detection and semantic artifacts. Furthermore, it will explain the differences between this paper and the existing research, as well as the main contributions of this work.

2.1 Image generation models

In recent years, image generation models have seen a large increase in complexity, popularity and relevance. They also have seen various shifts in architecture, each bringing a new level of quality and nuance to the images they can generate. One of the earlier models is the Variational Auto-Encoder [17], which encodes training data into a latent distribution, from which later random samples can be taken that are decoded to images. Its main follow-up is the Generative Adversarial Network (GAN) [18], which works by training two different models at the same time: the discriminator and the generator. The discriminator is passed both generated and real images and aims to label them correctly. The generator aims to generate realistic images which the discriminator cannot distinguish from real images. By having this ongoing "duel", hence the name adversarial, the two models improve one another. The final trained generator can be used to generate realistic images.

Another noteworthy type of model is the auto-regressive generative model [19]. It creates images by predicting the probability distribution of the next pixel, based on the previously generated pixels. It generates the pixels one by one, until a full image is generated. One of the more recent and prominent image generation models is the (latent) Diffusion Model [20, 12]. This type of model is trained on images with increasing levels of random noise added to them. The aim is to predict the level of noise that was added and subsequently remove it. After training, the model is able to start off with random noise and iteratively denoise it until a realistic image appears. Moreover, it is possible to condition this model on text embeddings, enabling text-to-image generation [12].

These differences in generative model architectures lead to distinct artifact patterns in their outputs, which detection methods can exploit but which also contribute to challenges in generalization across models.

2.2 Detecting synthetic images

Recent years have seen a growing body of research on the analysis and detection of AI-generated images. As generative models become more complex, the detection models, too, have to become more sophisticated.

Early works mainly focused on detecting images generated by GANs, by analyzing the frequency spectra of the generated images [7, 21]. These studies show that during the image generation process, an up-sampling operation is used which leads to detectable artifacts in the frequency domain of the generated images. By transforming the images into a frequency representation, the authors are able to respectively train a ResNet34 and a Linear Regression model that accurately distinguish real from generated images. Furthermore, Wang et al. [9] show that training a classifier (ResNet50) on a dataset which includes only samples from one type of CNN-based generator (ProGAN) results in a model which generalizes well to images from other CNN-based generators. This is because different CNN-based generators often share common artifacts from, for example, the up-sampling operations mentioned earlier. The authors also

demonstrate that aspects such as dataset diversity, data augmentation and data preprocessing during training contribute significantly to the ability to distinguish real from generated images. This is mainly because these operations manifest themselves as artifacts in the frequency spectrum, making it easier for a model to learn detectable patterns.

In addition, a study by Zhao et al. [22] focuses on detecting deepfakes by developing a multi-attentional network. The multi-attention enables the model to focus on several small local regions of, in this case, the face instead of extracting merely the global features. The model is able to detect deepfakes more accurately and also generalize better to other datasets. Furthermore, the study shows that the model pays specific attention to regions in the image which are often modified in deepfakes, such as the mouth and eyes. This claim that focusing on local patches instead of full images can increase the performance of a classifier is further supported in a study by Chai et al. [23]. The authors show that by limiting the receptive field of the classification model, it is forced to focus on small local cues in the image to classify it as either real or generated. This approach leads to increased generalization across different datasets and model architectures.

With the rise of (latent) Diffusion Models (DM), research moved towards detecting images generated by this type of model and answering the question whether previous detection models that worked on GANs also work on images generated by DMs. Corvi et al. [11] drew the conclusion that existing models that are trained on images generated by GANs do not perform well on DM-generated images. Although including data from one type of diffusion model in the training set helps the model to detect images generated by that type of model, it still does not generalize well to images from other types of diffusion models. These conclusions are backed up by Ricker et al. [8], who, in addition, prove that detection models trained on DM-generated images work well on GAN-generated images, but not the other way around. They suggest that this is due to the up-sampling artifacts that are an inherent part of GANs but not of DMs. Ojha et al. [13] aim to build a universal detector that generalizes across different types of generative models. They make use of a pretrained feature space of a large vision-language model. When classifying a new image, the image is mapped to this feature space and techniques such as nearest neighbor and linear probing are used to classify the image as real or generated.

These approaches mainly focus on generalizing across different types of model architectures, such as GANs and Diffusion Models. However, the generalization of classification models to images generated by the same model but with different types of semantics lacks proper investigation.

2.3 Bias in training datasets

In this research, we are dealing with a dataset bias challenge. Therefore, several previous studies on dataset bias are worth mentioning. Dataset bias in the computer vision domain was investigated as early as 2011 by Torralba et al. [24]. The authors investigate to what extent image datasets can contain biases and how these affect trained models and their ability to generalize to other datasets. In this study, an experiment is performed in which a linear SVM classifier is trained on a dataset consisting of subsets from 12 different datasets. The goal for the trained model is to be able to classify new images correctly in terms of which of the 12 datasets the image originates from. The results show an accuracy of 39%, which is significantly higher compared to an expected accuracy under random guessing of about 8%. Furthermore, an experiment is performed in which several object detection models are trained on one dataset and tested on others. Here, all datasets contain the same two image categories "car" and "person". The results show a significant performance drop when a model is tested on different datasets, indicating the existence of dataset bias.

Based on the aforementioned research, a follow-up study was performed in 2025 by Liu et al. [25], revisiting the same type of dataset classification experiment over a decade later with modern datasets and neural networks. The results still show an accuracy of 87.4% on three-way classification, indicating the datasets contain some clear form of pattern or bias. Furthermore, it is shown that the classification accuracy benefits from data augmentation and more samples. According to the authors, this suggests that "...the neural network attempts to discover dataset-specific patterns—a form of bias–to solve the dataset classification task." [25, p.2]

Although these studies establish the notion of dataset-specific patterns on which classifiers can overfit during training, it remains unclear where exactly this type of bias comes from in a dataset and how it can be mitigated.

2.4 Semantic Artifacts

One of the more recent developments in the field of AI-generated image detection is the analysis and mitigation of semantic artifacts. As mentioned in the introduction, semantic artifacts can induce a form of content bias in the training process, hurting generalization of the trained model. Zheng et al. [16] focus on "breaking" these semantic artifacts by implementing a pipeline based on patch shuffling. Each image is first divided into patches of equal size, which are then randomly shuffled. Then, each patch is sequentially passed through a single CNN to extract a set of features from it. Finally, the features are combined into a single vector that is fed into a linear classifier. The main goal of this technique is to break the global structure of each image. This encourages the final classification model to rely on the local features within each patch to classify an image as real or generated, thereby reducing overfitting on global structure and semantics. When compared to state-of-the-art methods, the study shows an improvement of 10.59% in average accuracy across all test sets. Furthermore, the authors show the effects of patch shuffling on semantic artifacts in a separate experiment. Here, two models are trained separately on different datasets, with each dataset containing images of only one category of object or scene. When each model is tested on a test set of the other dataset, accuracy is near random. But, when training each model on a patch-shuffled version of their respective training data, test accuracy on the other datasets increases significantly for both models.

The same idea of breaking semantic artifacts by splitting the image into patches was studied in two papers [26, 27]. Both papers showed an increase in cross-dataset performance when focusing on patches of images or patch-shuffled versions of the images during training. In [28], the authors aim to improve on the methods described above by proposing Semantic-Agnostic Fine-tuning (SemAnti). A CLIP model is used in which the layers that are shown to capture the semantic content of the images are frozen during fine-tuning. During fine-tuning, the model gets as input patch-shuffled images, such that only the layers that are sensitive to, for example, generator artifacts are updated. They achieve state-of-the-art performance on several benchmark datasets such as the AIGCDetectBenchmark [29] as of their publication.

These studies have introduced ways to mitigate semantic artifacts and improve model generalization. However, a proper understanding and way of measuring semantic artifacts remains uncovered. Furthermore, their impact on model generalization is only addressed in one small experiment, leaving room for further investigation.

3 Methodology

This section will go into further detail on the methods used to answer the main research questions. More specifically, it goes over the process of examining the relationship between the level of semantic artifacts in a dataset and the generalization of a model trained on that dataset. As mentioned in section 1, this means we first aim to describe the level of semantic artifacts in a dataset using a quantitative method. We then analyze whether AI image classification models trained on datasets with different levels of semantic bias differ in their ability to generalize to other datasets. By following this method, the effects of patch shuffling on semantic artifacts and model performance can also be evaluated.

3.1 Overall method

The overall method can be described as follows: To start, several datasets are created with an equal amount of real and AI generated images. For each of these datasets, the level of semantic artifacts is described using the noise residual power spectra of both classes of the dataset. Of these two spectra, the difference in angular variance is calculated, which is used as the main indicator for the level of semantic artifacts in the dataset. The final metric that is introduced is referred to as the Semantic Artifact score or **SEMAR** score. Next, a classification model is trained on each dataset and subsequently evaluated on a test set from the same dataset as well as on test sets from all other datasets. The same steps of training and evaluating are repeated using an additional preprocessing step of patch shuffling the input images. The SEMAR scores and evaluation results are used to establish a conclusion on the measurability of semantic artifacts in a dataset, its correlation to model generalization and the effects of patch shuffling on the level of artifacts and model generalization.

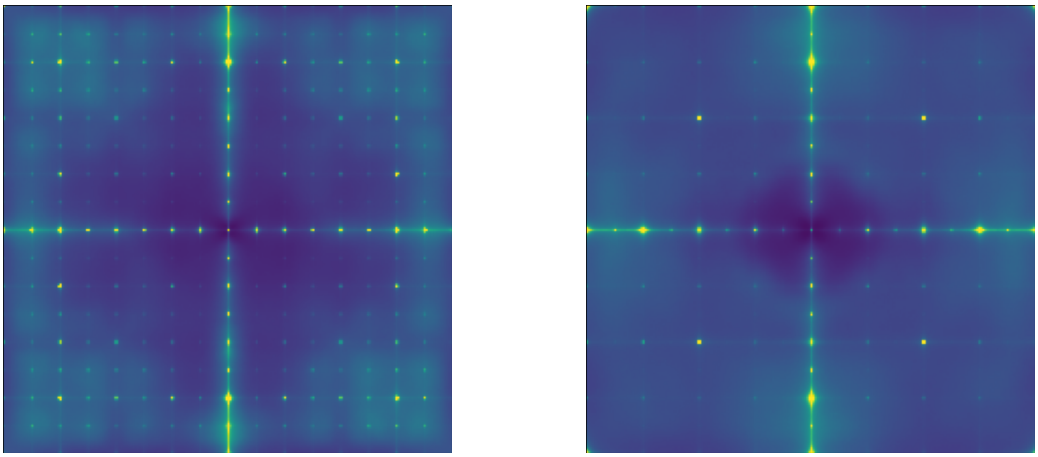
3.2 Establishing the Semantic Artifact (SEMAR) score

This section describes the main pipeline for establishing the level of semantic artifacts in a dataset. The pipeline takes as input (part of) a dataset and outputs the SEMAR score. The ability to establish a SEMAR score for a dataset will allow us to actively measure the level of semantic bias in a dataset, which in turn will enable experiments such as analyzing its correlation with model generalization. The pipeline consists of three main components, which are: (1) calculating a noise residual power spectrum, (2) deriving the angular variance from this spectrum, and (3) using the angular variance to compute the SEMAR score of the dataset. The first two subsections will describe the idea behind using noise residual power spectra and angular variance. The next two subsections will go into further detail on the technical implementation of these components. The final subsection will describe how everything comes together in order to calculate the final SEMAR score. Moreover, a formal mathematical definition of the SEMAR score will be given.

3.2.1 Noise residual power spectra and Artifacts

A noise residual power spectrum aims to give a visual representation of how the energy (power) is distributed across spatial frequencies in an image or set of images. It can be seen as a type of "fingerprint" that captures the underlying patterns of an image or dataset [11]. Since the spectrum is computed from noise residuals rather than the original image content, it emphasizes subtle frequency components that may otherwise be invisible in the image content [8, 30]. In practice, this means that different AI image generation models can be distinguished by analyzing the noise residual power spectrum of their generated images [31]. These may contain certain patterns like a grid or regularly spaced frequency peaks, which are an artifact of the

underlying generation process from the latent diffusion model. These artifacts are commonly referred to as generator artifacts [16]. An example of generator artifacts can be found in the spectra shown in image 1.



(a) LSUN-church, generated dataset

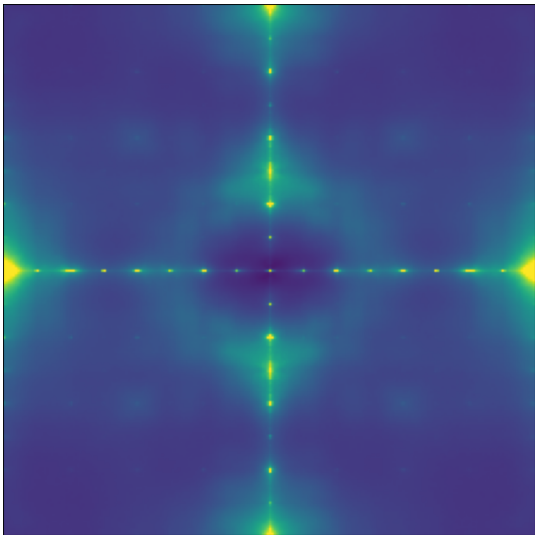
(b) LSUN-bedroom, generated dataset

Figure 1: Different levels of generator artifacts can be observed in these two noise residual power spectra of different datasets, each dataset consists of 5000 unique images.

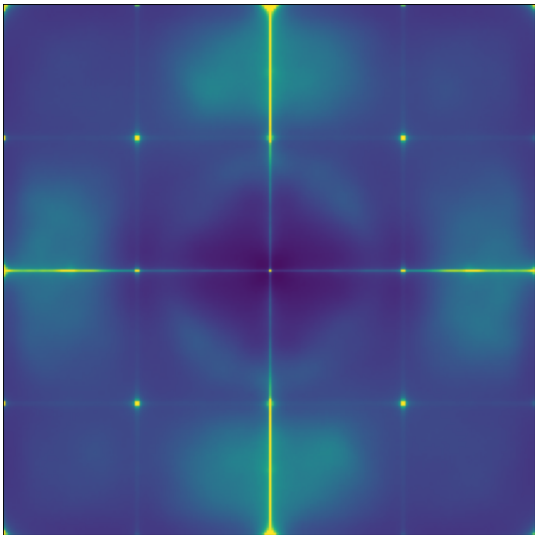
Semantic artifacts are also visible in the noise residual power spectrum, but in a different way. Since semantic artifacts are related to the actual semantic content of the images, semantic artifacts generally appear as bigger shapes or circle-like patterns in the spectra. This can represent aspects such as the dominant orientation, shape and object size in a dataset [16]. The distribution of energy across the frequencies in the spectrum corresponds to a large extent with the homogeneity of the dataset. Energy in the center of the spectrum corresponds to dominant low frequencies, which inherit the general shape instead of per-pixel differences in the images. Energy in the outer edges of the spectrum corresponds to dominant high frequencies and more fine details [32]. Therefore, spectra of datasets with a higher level of homogeneity generally tend to have more energy in the low-frequency areas of the spectrum, while the spectra of less homogeneous datasets tend to show more energy in the high-frequency areas and a more even distribution of energy across the spectrum.

An example can be found in Figure 2, which shows two noise residual power spectra of different datasets. The first plot is based on the real image subset of the CelebA dataset [33]. It shows a clear round shape in the center, likely due to the homogeneous nature of the dataset, which contains only human faces from the same angle [16]. The second plot in image 2 is based on the generated half of the ImageNet dataset. The ImageNet dataset contains an equal amount of samples from 1000 different classes. It can be observed that this spectrum, unlike the one from CelebA, contains regularly spaced frequency peaks and a grid pattern. This indicates the spectrum is created based on AI generated images instead of real images. Furthermore, compared to CelebA, this spectrum shows more energy on the outer edges of the spectrum. Since ImageNet contains many different object classes, shapes and textures, the resulting spectrum distributes the energy more evenly across the different frequencies.

Furthermore, although artifacts from preprocessing steps are not considered semantic artifacts in this research, these types of artifacts are still visible in several areas of the frequency power spectrum. Moreover, some of these areas overlap with the area in which semantic artifacts exist. As can be seen in Figure 2, both spectra show a clear cross-like pattern, consisting of high-frequency peaks. While existing literature does not explicitly explain this last type of artifact, the experiments in this research have demonstrated that it originates from and is enhanced by JPEG compression [30]. However, since exactly the same preprocessing steps are used for each dataset, it is expected that these preprocessing artifacts do not have any influence on the relative differences in SEMAR scores between the datasets. Further analysis of the spectra from the datasets used in this research will be provided in section 4.



(a) CelebA, non-generated dataset



(b) ImageNet, generated dataset



(c) Samples from the CelebA dataset



(d) Samples from the generated ImageNet dataset

Figure 2: Top row: Different types of semantic artifacts can be observed in these two noise residual power spectra of different datasets, each dataset consists of 5000 unique images. Bottom row: Samples of each dataset to demonstrate the level of homogeneity between images.

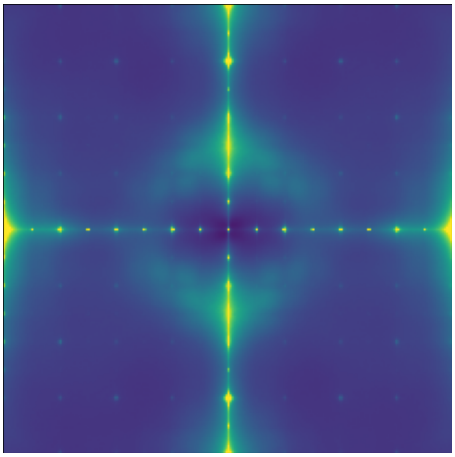
3.2.2 Angular variance

The previous section discussed the different types of artifacts and where these are visible in the frequency power spectrum. It was observed that semantic artifacts generally appear as bigger shapes or circle-like patterns in the center of a spectrum. The next step is to be able to use this observation in order to measure the level of semantic artifacts in a dataset through its frequency power spectrum.

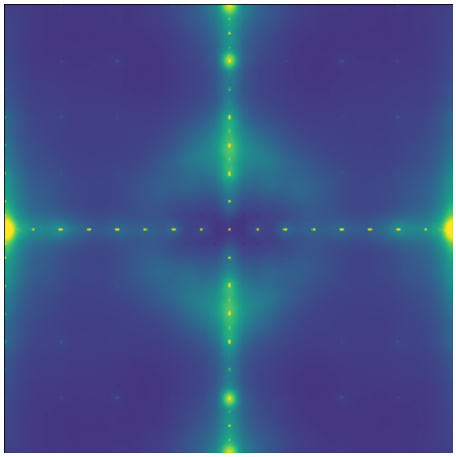
Semantic artifacts manifest themselves as a specific pattern visible in the frequency power spectrum of a dataset [16]. The study by Zheng et al. [16] introduces a technique called patch shuffling, which is expected to reduce or "break" the semantic artifacts at least partially, likely improving generalization of a model trained on that dataset [16]. Patch shuffling means the image is divided into patches of a certain size and then randomly shuffled, causing the original semantics of the image to disappear while the underlying artifacts related to the generative models or preprocessing steps remain [16].

The paper by Zheng et al. [16] presents an experiment where the reduction of semantic artifacts in the frequency power spectrum through patch shuffling is visible. The results show that the patch-shuffled spectra clearly exhibit a more even, smoother distribution of energy. Large peaks are smoothed out, and there appear to be fewer significant artifacts in the center. A similar experiment was performed in this research, and an illustration of the effects of patch shuffling on the noise residual power spectra can be found in Figure 3. The figure shows the noise residual power spectra of the same dataset with and without patch shuffling applied to the image data. It can be observed that the overall spectrum is smoother and has less contrast, especially when looking at the shape in the center of the spectrum. Furthermore, the intensity of the individual frequency peaks seems to be lower.

Based on the idea that patch shuffling reduces semantic artifacts and that this reduction manifests itself in a smoother and less intense area of the frequency power spectrum, measuring the difference in this area of the spectrum before and after applying patch shuffling is considered as a way of measuring the level of semantic artifacts in a dataset. Initially, two different ways of measuring the difference between the shuffled and non-shuffled spectrum were considered. The first way is to measure the total difference in energy in the specific area of the spectrum where semantic artifacts manifest themselves, before and after applying patch shuffling. However, Parseval's theorem states that the total energy in the spatial domain always equals the total energy in the frequency domain [34]. Since patch shuffling the image does not change the actual pixel values in the spatial domain, the total energy in the spectra also does not change. What does change, however, is how the energy is distributed when comparing the shuffled and non-shuffled spectra. Therefore, the difference in angular variance between the two spectra is used to measure the level of semantic artifacts. Angular variance first measures the mean energy within a bin at a given radius from the center of the spectrum. It then applies the same calculation to all bins in a 360°rotation around the center. With this, the variance of energy across all angles can be compared. This method assumes that the level of semantic artifacts in a dataset correlates directly to the decrease in angular variance of the noise residual power spectrum when applying patch shuffling. Although this assumption can be validated to a certain extent based on earlier papers such as [16], the results of this research must further validate this assumption.



(a) LSUN-church, non-generated dataset without patch shuffling



(b) LSUN-church, non-generated dataset with patch shuffling

Figure 3: The effects of patch shuffling can be observed in these two noise residual power spectra of the same dataset, the dataset consists of 5000 unique images.

3.2.3 Pipeline - Creating the power spectrum

The first main component in computing the SEMAR score is a pipeline for creating the noise residual power spectrum of a dataset. This pipeline is based on the existing literature on analyzing artifacts of datasets [16, 30]. These papers use the same steps to create the noise residual power spectra, and the same procedure will be used in this research. The process is explained step by step below. Furthermore, an illustration of the intermediate steps can be found in Figure 4.

1. A set of images is selected as input. In this research, the input dataset always consists of one specific half of a dataset, which is either the real or the generated half.
2. Since preprocessing could alter the level of artifacts in the image, exactly the same preprocessing steps are performed on the image data as for training the classification model. Preprocessing steps are described in more detail in section 3.3.2, and mainly contain resizing operations.
3. When calculating the spectrum on patch-shuffled data, patch shuffling is applied at this point in the process. The image is divided into patches, the patches are randomly shuffled and then reassembled into a single image.
4. A denoising filter is used to extract the noise residual of each image [35]. The selected denoising filter is used in similar research involving the same type of experiments, ensuring possible comparability to earlier work [16, 30].
5. For each image, a Fourier transform is applied, after which the corresponding power spectrum of the image is generated by taking the square of the transform.
6. Finally, the average is taken over all power spectra, resulting in the average noise residual power spectrum of the dataset.

3.2.4 Pipeline - Calculating angular variance

The second component is a pipeline for computing the angular variance from a noise residual power spectrum. This subsection describes this pipeline step-by-step.

1. The center o of the spectrum is determined.
2. For each pixel in the spectrum, the distance to the center r and the angle θ are calculated.
3. Based on the values of each pixel, a radial mask selects all pixels that fall within a specific range from the center o of the spectrum. This results in capturing the values in a circle shape with its center being the center of the spectrum. In this research, a radius of $r = 60$ pixels is used since this is expected to capture most of the frequencies related to semantic artifacts in the frequency power spectrum.
4. The circular region is divided into 360 different bins, each corresponding to $1°$of the circle. This aims for a tradeoff between accurately capturing the subtle changes in energy in the spectrum and computability. The mean energy of each bin is then computed by taking the average of all values within that bin.
5. Finally, the angular variance is computed as the variance of the 360 mean bin energies.

3.2.5 Pipeline - Calculating the SEMAR score

Using the pipelines for computing the noise residual power spectra and the angular variance, the final SEMAR score for a single dataset is calculated as follows:

1. The noise residual power spectra are computed for both the real- and generated class of the dataset, once using patch-shuffled images and once using non-shuffled images. This results in four different spectra R, R_s, G, G_s, where G_s denotes the generated class using patch shuffling and R denotes the real class without patch shuffling.
2. All spectra are first normalized, in order to enable a fair and consistent comparison between scores of different datasets.
3. For each normalized spectrum, the angular variance is calculated.
4. The relative change in angular variance between the shuffled and non-shuffled spectra of each class is calculated. To illustrate, the relative change for the real class is calculated by $(Av(R) - Av(R_s))/Av(R)$, where $Av(\cdot)$ denotes the angular variance of the spectrum. Here, the change is defined as the decrease in angular variance after patch shuffling, hence $Av(R_s)$ is subtracted from $Av(R)$ and not the other way around.
5. This results in two values, both representing the relative change in angular variance for the real and generated class.
6. The final SEMAR score for the dataset is calculated by taking the average of the two values from the previous step.

Based on the previous steps, the SEMAR score can be mathematically defined as follows:

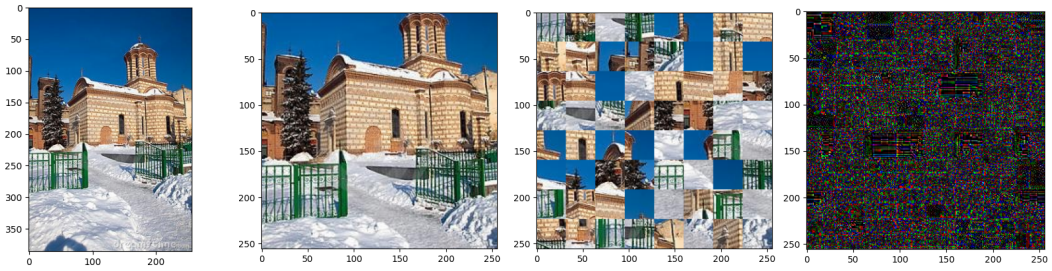
Let $\mathcal{D} = \{\mathcal{R}, \mathcal{G}\}$ be a dataset consisting of a real class $\mathcal{R}$ and a generated class $\mathcal{G}$, with $\mathcal{C} \in \{\mathcal{R}, \mathcal{G}\}$. The relative change in angular variance for a class in $\mathcal{C}$ due to patch shuffling is defined as:

$$\Delta_{\mathcal{C}} = \frac{Av(\mathcal{C}) - Av(\mathcal{C}_s)}{Av(\mathcal{C})} \tag{1}$$

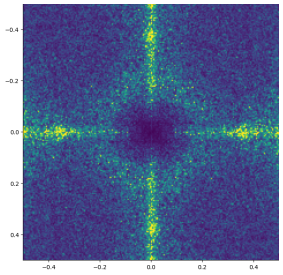
where $Av(\mathcal{C})$ and $Av(\mathcal{C}_s)$ denote the angular variance of the noise residual power spectrum of class $\mathcal{C}$ without and with patch shuffling, respectively.

The SEMAR score for dataset $\mathcal{D}$ is then defined as:

$$\text{SEMAR}(\mathcal{D}) = \frac{1}{2}\left(\Delta_{\mathcal{R}} + \Delta_{\mathcal{G}}\right) \tag{2}$$



(a) Original (b) Resized (c) Shuffled (d) Denoised



(e) Power spectrum

Figure 4: The pipeline for creating the power spectrum of a single image. When performed for all images of the input dataset, the average of these can be taken. The result of this will be the noise residual power spectrum that represents the whole input dataset.

3.3 Detecting generated images

3.3.1 Data generation

The AI-generated half of each dataset is created using a pretrained LDM. For each dataset, two important parameters were used in the model's generation process. The first is the Denoising Diffusion Implicit Models (DDIM) sampler [36] and its number of sampling steps, which controls the number of denoising steps that are used in the generation process. In short, fewer steps result in a faster generation process but lower-quality images. Conversely, a higher step size results in higher quality but slower generation. For each dataset, a number of steps is chosen that achieves a tradeoff between quality and speed. The second parameter η controls the amount of stochastic noise injected during the DDIM sampling process. Therefore, $\eta = 0$ is used to ensure deterministic sampling and a fair comparison of artifacts across datasets.

3.3.2 Preprocessing

To ensure a fair comparison between different datasets and their level of semantic artifacts, it is important to maintain a consistent level of preprocessing. This ensures that the content of the data (i.e., the image) itself is the only variable that changes, which allows us to answer the research questions with more confidence. Moreover, since the purpose of this research is mainly to investigate the impact of artifacts arising from the semantic and visual content of the images, preprocessing steps are kept to a minimum.

First, all images are ensured to have the same resolution of 256 x 256 pixels. Generated images are generated in these dimensions already. For the non-generated images, dimensions vary. To handle this, all images are first resized using bilinear interpolation, such that at least one side matches the final expected length of 256 pixels. The other side is adjusted proportionally. For square images, this already scales the dimensions to 256 x 256 pixels. For non-square images, a cropping operation is performed that takes the center of the image and crops the image equally on both sides until the desired dimension is reached.

Next, an optional preprocessing step applies patch shuffling to the image. This concept is based on the paper by Zheng et al. [16], and divides the image into patches of a specific size, randomly shuffles them and then reassembles them. The patch size is set to 32 x 32 pixels, resulting in an image divided into 64 different patches without any padding or margin around them. This choice of patch size aims to break the semantic structure of the images, while maintaining the local patterns within the patches. Furthermore, an ablation study in [16] shows that a patch size of 32 x 32 pixels results in the best performance of the trained classifier models. For each dataset, a model is trained with and without patch shuffling in order to analyze its effects on the generalization of the classifier. The reasoning behind this process is provided in section 3.2 and lies beyond the purpose of this section.

Other preprocessing steps were investigated as well, such as extra JPEG-compression, augmentation of the images and random cropping. These steps were excluded from the pipeline, since they are expected to merely add additional noise or preprocessing artifacts to the datasets. Although the effects of this could be valuable for future research, it lies beyond the scope of this work.
Finally, each image is converted to a PyTorch [37] tensor to enable processing by the neural network.

3.3.3 Training the classifiers

For each dataset, a ResNet18 classifier model is trained, evaluated and tested from scratch. Initially, a ResNet50 classifier was used, but this resulted in a model more prone to overfitting on the training set, likely due to its larger number of parameters. For the ResNet18 model, binary cross-entropy with logits loss is used as the loss function.

Next to the set of models described above, a second set of models is trained, evaluated and tested in exactly the same way on a patch-shuffled version of the data. Patch shuffling is applied during runtime, and is applied to the training set, evaluation set and test sets. This enables an analysis of the possible effects that patch shuffling has on the generalization of models.

4 Experiments and results

This section will go over the conducted experiments and the results obtained using the previously described method. First, the details on the datasets, training and evaluation will be described. The section then moves on to three main parts, each of which focuses on one of the research questions with its associated experiment and results. Furthermore, the results from each experiment will be discussed and interpreted at the end of each section. Finally, an additional experiment is performed which examines the influence and impact of JPEG compression on the SEMAR score of a dataset. Moreover, the generalization of classifiers trained on datasets with and without JPEG compression is analyzed.

4.1 Datasets

4.1.1 Dataset selection and construction

The reliability and clarity of the results depend to a large extent on the datasets used in the experiments. The most important factor to consider when selecting datasets is the range of semantic variation, i.e., the extent to which datasets differ in aspects such as the number of objects, scenes, lighting conditions and preprocessing. Ideally, the selected datasets should span a wide range of semantic variation. With, at one end, datasets with very little variation in the aforementioned aspects, and at the other end, datasets with more substantial variation. This ensures that the set of datasets covers a wide range of semantic artifact levels, possibly making it easier to highlight the differences between them.

A total of five datasets are used in this research. Since each dataset is evaluated on all other datasets as well as on itself, this yields sufficient data points to draw reliable conclusions. For each dataset, a sample of 5,000 images is collected for the non-generated class. An additional 5,000 images are obtained by using a pretrained latent-diffusion model. With a total of 10,000 images per dataset, this is expected to provide a sufficient training set for a ResNet-18 classifier model.

4.1.2 Datasets

The datasets used in this research are: ImageNet [38], CelebA [33], FFHQ [39], LSUN-Church and LSUN-Bedroom [40]. These datasets are expected to cover a wide range of semantic variation and therefore a wide range of semantic artifact levels. CelebA on one end has only one type of image (faces of celebrities) and mostly similar angles, cropping and lighting. ImageNet, on the other hand, is constructed by sampling ten images: five real and five generated from each of the 1000 classes. This results in high variation in not only the classes but also in compositions, angles and colors. The FFHQ and LSUN datasets are expected to lie somewhere in between the two aforementioned datasets in terms of semantic variation, with FFHQ also consisting of images of human faces, but having "vastly more variation than CelebA in terms of age, ethnicity and image background ..." [39, p.8]. The two LSUN datasets contain images of churches and bedrooms, respectively, in different settings, lighting and angles. Further details on the specific attributes of each dataset can be found in Table 6 in the Appendix.

Additionally, the study by Zheng et al. [16] on breaking semantic artifacts uses the same datasets to establish an understanding of semantic artifacts and baseline results on generalization of classifiers trained on these models. This enables the comparison of semantic artifact analyses between this study and theirs.

4.1.3 Dataset generation details

The pretrained LDMs are obtained from the official CompVis repository[1], maintained by the original authors of the LDM paper [12]. This repository provides models fully pretrained (not fine-tuned) on each of the five aforementioned datasets, as well as the necessary architecture and code to run them. The repository also provides a set of autoencoder models pretrained on the OpenImages dataset [41]. These models are used in combination with the LDMs. The autoencoder models used in this research are in line with the models recommended by the authors of the CompVis repository, and are all specified in the Appendix A.2.



(a) CelebA (b) FFHQ (c) ImageNet (d) LSUN-Bedroom (e) LSUN-Church

Figure 5: Samples from the five datasets. Each plot shows four example images illustrating dataset characteristics.

4.2 Implementation details

4.2.1 Training details

The same implementation details are used for all experiments. This includes an 80/10/10 dataset split for training, evaluation and testing respectively. Furthermore, every model is trained for 20 epochs, in each of which the entire train set of 8,000 images is passed through the network in batches of 32 images. This results in a total of 250 network updates per epoch. After each epoch, the model's accuracy and Average Precision (AP) are measured on the evaluation set. Early stopping is applied if the AP does not improve for five consecutive epochs. This aims to reduce the risk of overfitting on the training data. Moreover, model parameters are optimized using the Adam optimizer [42] with a learning rate of 0.0003 and a β_1 value of 0.9, which aims to provide stable convergence. Lastly, all evaluation runs are repeated three times to ensure that no large variance in performance exists between them.

All of the experiments in this research were performed using the ALICE compute resources provided by Leiden University. Specifically, experiments were run on the gpu-l4-24g partition, which consists of 8 compute nodes equipped with 4 NVIDIA L4 GPUs (24 GB VRAM) per node.

4.2.2 Evaluation

In order to draw conclusions from the performed experiments, two components are needed. The first component is the SEMAR score of each dataset, as discussed in section 3.2. The second component is a cross-evaluation of each trained model on the different test splits of all datasets using the AP and accuracy scores. Since there are five datasets in total, this results in 5×5 evaluations yielding a total of 25 data points. Using the SEMAR scores and cross-testing results, possible correlations between the two can be analyzed.

[1] https://github.com/CompVis/latent-diffusion

4.3 A quantitative measurement for semantic artifacts

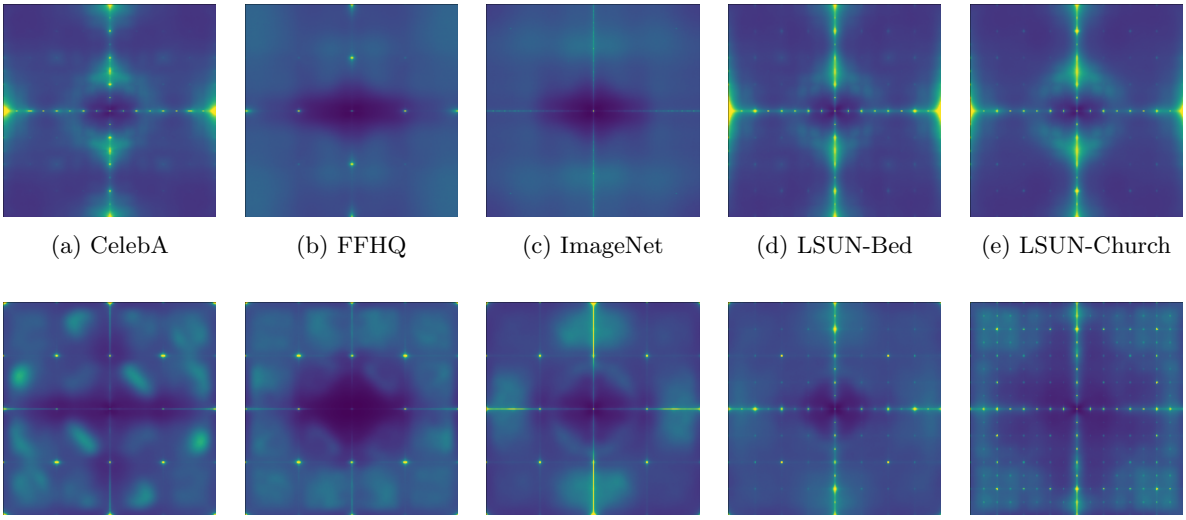
The first experiment aims to answer the research question of whether semantic artifacts in a dataset can be described using a quantitative method. This intends to give a better understanding of what semantic artifacts are and how they relate to different types of datasets. These findings can, in turn, be used to support the outcomes of subsequent experiments on semantic artifacts and model generalization.

The method used in this experiment is described in section 3.2. For each dataset, a noise residual power spectrum is created for each class. Using the difference in angular variance between the two spectra, the final SEMAR score is computed to describe the semantic artifacts in the entire dataset.

4.3.1 Results

Figure 6 shows the noise residual power spectra for each of the two classes in the used datasets. Furthermore, Table 1 includes the SEMAR scores for each of the datasets. These results contribute to answering the research question in two main ways. First, the spectra can be analyzed to identify differences in semantic artifact patterns across datasets and between the two classes within each dataset. The analysis of the spectra can be compared to the expected level of artifacts in each dataset based on its characteristics, such as the level of homogeneity. This aims to provide insight into whether the spectra offer a meaningful representation of the level of semantic artifacts in a dataset. It is, however, important to note that there does not exist a ground truth for the definition of semantic artifacts in a dataset and their manifestation in the frequency power spectrum. Therefore, the results of this experiment cannot be fully validated for correctness but merely compared to expected results based on previous studies[16].

Second, the dataset's spectra can be compared with their corresponding SEMAR scores. This comparison makes it possible to evaluate whether the SEMAR score aligns with the expected artifact levels based on dataset characteristics and the artifacts visually apparent in the spectra.



(a) CelebA (b) FFHQ (c) ImageNet (d) LSUN-Bed (e) LSUN-Church

(f) CelebA (g) FFHQ (h) ImageNet (i) LSUN-Bed (j) LSUN-Church

Figure 6: Power spectra of each of the datasets' classes: Real (top row) and generated (bottom row).

4.3.2 Discussion

Figure 6 shows a clear separation within the spectra of the real classes across datasets. CelebA, LSUN-Bed and LSUN-Church show a similar pattern, with a similar central shape and a clearly defined cross throughout the spectrum. FFHQ and ImageNet, on the other hand, share different characteristics and do not exhibit the same types of artifacts as the previously mentioned datasets. It can be observed that FFHQ and ImageNet show a more even spectrum with less significant peaks at certain frequencies. This is expected due to the higher level of heterogeneity that FFHQ and ImageNet have. Unlike CelebA and the LSUN datasets, these datasets contain either a large variety of different objects (ImageNet) or a large variety of angles and compositions of the same object (FFHQ).

Furthermore, the circular shape in the center of the spectra of the real classes of CelebA and the LSUN datasets is likely an artifact of the higher level of homogeneity of these datasets. As mentioned in Section 3.2, the center of the power spectrum corresponds to the lower frequencies of the dataset's images that represent large shapes, objects and overall composition. Therefore, it is expected that the shared type of artifact between the CelebA and LSUN datasets is due to their common characteristic of having mostly the same type of object in a similar orientation and setting.

In addition, all spectra of the real classes of the datasets show a cross throughout the center of the spectrum of different intensities. This is likely an artifact of the JPEG compression that is performed on the images in the datasets. The different levels of this artifact between, for example, ImageNet and CelebA could be due to differences in JPEG compression. Although every dataset is preprocessed in exactly the same way in the experiment, preprocessing steps performed prior to obtaining the datasets, such as different levels of JPEG compression, cannot be ruled out. Furthermore, during experimentation, it has been observed that increasing the JPEG compression slightly increases the intensity of this type of artifact in the spectra. This provides additional support for the different intensities of this artifact.

Comparing the spectra of the generated classes of the datasets in Figure 6 to the spectra of the real classes, a clear type of artifact stands out. These are the regularly spaced frequency peaks, which are artifacts of the LDM generation process as described in Section 3.2. This artifact seems most present in the spectrum of the LSUN-Church dataset, likely due to the different type of autoencoder used to generate its images. Interestingly, the spectra of the generated classes contain more energy in the area outside the center. For some datasets, such as FFHQ and ImageNet, a possible explanation is that the LDM is biased towards only generating a specific subset of the dataset it has been trained on. This can be observed when comparing the real and generated spectra of these datasets: Regions that already contain some energy in the spectrum of the real class, contain more energy in the spectrum of the generated class.

Moreover, a general observation can be made that the similarity between the spectra of real and generated images decreases when moving toward the higher frequency regions of the spectrum. Whereas in the center of the spectra, artifacts of the two classes seem more similar. The outer edges of the spectra show more significant differences. The same observation was made in an ablation study in [16].

Looking at the SEMAR scores in Table 1, the two LSUN datasets clearly stand out as having the most (semantic) artifacts in the dataset as a whole. This corresponds to the intensity and the types of the artifacts visible in their spectra. It can also be explained by the characteristics of the datasets, as these contain images of only one specific object. CelebA has a similar level of homogeneity as the LSUN datasets. Its lower SEMAR score could be explained by the smaller number of generator artifacts in the generated class. FFHQ and ImageNet have the lowest SEMAR scores of all datasets. This corresponds to the artifacts that can be observed in their spectra, as well as with the characteristics of the datasets themselves.

To conclude, frequency power spectra clearly reflect specific characteristics of the dataset from which they are derived. The level and types of artifacts in each dataset seem to also partially correlate with their SEMAR scores. By analyzing frequency power spectra, clear differences between real and generated images can be observed. Generated images can be distinguished by regularly spaced frequency peaks, which are artifacts of the LDM generation process. These artifacts vary depending on the type of autoencoder that is used in the generation process. Furthermore, datasets with low homogeneity show a smoother and more even power spectrum than datasets with high homogeneity. Datasets with high homogeneity, on the other hand, clearly show more energy in the center of the spectrum. Finally, JPEG compression plays a significant role in the frequency power spectrum. This type of compression manifests itself as a clearly defined cross across the entire spectrum.

4.4 Semantic artifacts and model generalization

This experiment aims to show the relationship between the level of semantic artifacts in a dataset and the generalization of a model trained on that dataset. For this experiment, the five datasets mentioned in the previous section are used as well as their SEMAR scores. For each dataset, a model is trained and evaluated using the implementation details described in section 4.2.

4.4.1 Results

The results of this experiment are displayed below. Table 1 provides an overview of each of the datasets, its SEMAR score, the performance on a hold-out test set of the same dataset and its performance when evaluated on all other datasets. The last column denotes the generalization gap, defined as the difference between a model's performance on a hold-out test set from the dataset it was trained on and its performance on the other datasets. Moreover, Figure 7 shows a scatterplot including trend lines of the SEMAR score and generalization gap of each of the datasets measured in AP and accuracy. In addition, Table 2 shows a more detailed overview of the performance of each model. Each evaluation has been performed three times. The difference in performance between each run in both AP and accuracy did not exceed 0.5.

		Performance (self)		Performance (others, mean)		Generalization gap	
Dataset	SEMAR	AP	Acc.	AP	Acc.	AP	Acc.
CelebA	0.23	99.99	98.89	59.94	53.30	40.06	45.59
FFHQ	0.16	99.86	92.44	58.61	51.33	41.26	41.11
LSUN-Bed	0.27	95.72	81.75	49.84	51.99	45.88	29.76
LSUN-Church	0.28	99.43	90.73	56.62	53.66	42.81	37.08
ImageNet	0.18	96.35	81.96	48.35	46.83	48.00	35.14

Table 1: For each of the five datasets: the SEMAR score, performance on the same dataset and average performance on other datasets. The difference between the two performance metrics is denoted as the generalization gap.

	CelebA		FFHQ		LSUN-Bed		LSUN-Church		ImageNet		Average	
Dataset	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.
CelebA	99.99	98.89	78.8	52.62	55.35	53.83	59.37	57.46	46.22	49.29	67.95	62.42
FFHQ	75.67	51.51	99.86	92.44	58.38	51.71	46.72	50.50	53.65	51.61	66.86	59.55
LSUN-Bed	45.04	49.50	66.49	62.40	95.72	81.75	43.43	48.89	44.40	47.18	59.02	57.94
LSUN-Church	73.80	63.21	42.41	43.45	62.47	57.56	99.43	90.73	47.79	50.40	65.18	61.07
ImageNet	53.36	49.80	50.14	48.99	38.83	37.50	51.08	51.01	96.35	81.96	57.95	53.85
Average	69.57	62.58	67.54	59.98	62.15	56.47	60.01	59.72	57.68	56.09	–	–

Table 2: Cross-dataset performance of each of the five datasets. Each row indicates a model trained on a dataset. The columns indicate the different test performances on the datasets.

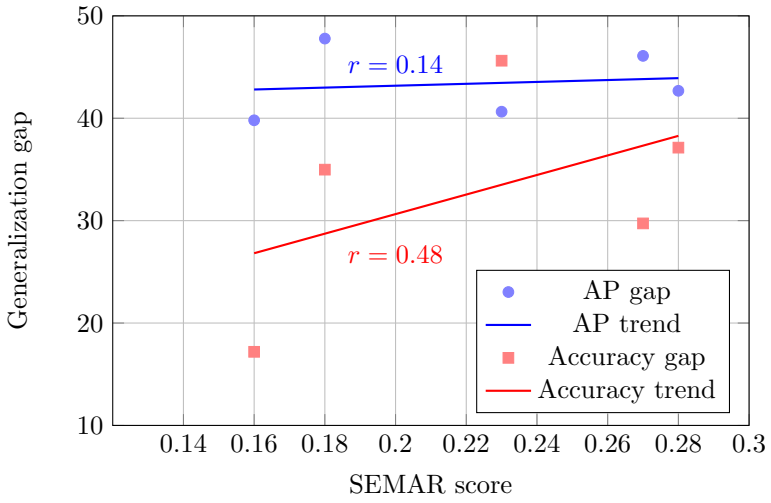


Figure 7: Relationship between SEMAR score and generalization gap measured in AP and accuracy. Both the data points and a trend line for each metric are displayed, as well as the Pearson correlation coefficient. The trend lines and Pearson correlation coefficients should be interpreted with caution given the small sample size.

4.4.2 Discussion

First, for all models, performance within the same dataset is high in terms of both the AP and accuracy. Table 1 shows that, while there is some variation between datasets in terms of accuracy, the AP for all models surpasses a value of 95 and accuracy is well above 80%.

Furthermore, Table 1 shows that the model trained on CelebA shows the best average performance on other datasets and the lowest generalization gap in terms of AP. An explanation for this could be that CelebA's real-class spectra share similar types of artifacts with the LSUN datasets, which could make it easier for CelebA to generalize to those datasets, thereby increasing its overall average performance. An additional reason for this could be the relatively high performance of the CelebA-trained model on FFHQ, which can be explained by the similarity in image types between the two datasets. Interestingly, CelebA also shows the highest performance when used as a test set. This means that, on average, other models can classify images from CelebA as real or generated most easily.

The model trained on ImageNet, on the other hand, shows the lowest performance on other datasets, indicating poor generalization. This does not seem to correlate with the relatively low SEMAR score, but may be due to the large number of different image classes it is trained on. This can lead to the model learning features that are too general, resulting in poor generalization. When looking at the performance of the ImageNet model in Table 2, this poor average performance can be mainly explained by its significantly lower performance on the LSUN-Bedroom dataset compared to other models. This lower performance could be due to the higher level of detail that the images in the LSUN-Bedroom dataset contain compared to other datasets. For a model that is trained on a large set of different image classes like ImageNet, it can be expected that this model relies more heavily on generic features of the images for classification. It

could therefore be more difficult to extract the local features that are needed to accurately classify images from a very specific domain such as the LSUN-Bedroom dataset. This finding could possibly provide counter-evidence for the existence of a relationship between the SEMAR score and the generalization of a classifier model. However, further experimentation on different types of datasets would be needed to construct a proper conclusion.

Looking at the average performance in Table 2, several trends can be observed. First, ImageNet has the lowest mean performance as test set, meaning this dataset is the hardest for other models to generalize to. This could be explained by the number of image classes this dataset contains compared to the other datasets. Moreover, CelebA and FFHQ show relatively high performance when evaluated on each other, compared to the other datasets. A possible explanation is that both datasets contain the same type of images, namely human faces. Furthermore, FFHQ seems to generalize better to CelebA than the other way around. This could be due to CelebA consisting of images from mostly the same angle and orientation, while FFHQ contains more diversity in terms of these aspects. These relatively high scores also contribute to the explanation of why CelebA and FFHQ show the highest average performance on all other datasets.

An additional observation can be made on the overall differences between AP and accuracy. In general, for each combination of training and test set, AP is slightly higher than the accuracy. This could indicate that, in general, models are able to distinguish between real and generated images. However, due to the classification threshold, this ranking ability is not always converted into correct predictions (see Section 3.2.2).

To conclude, in general, it can be observed that models perform well when they are tested on the same dataset. When tested on different datasets, the performance significantly decreases. A possible explanation for this could be that models tend to overfit on the different artifacts in the training data, including semantic artifacts, preventing them from generalizing well to other datasets without those types of artifacts. Moreover, based on Figure 7, there appears to be a slight correlation between the level of semantic artifacts in a dataset and the generalization of a model trained on that dataset. This correlation is especially apparent when measuring the generalization gap in accuracy rather than AP. This suggests that models tend to suffer larger accuracy drops than AP drops as the SEMAR score increases. Lastly, the low cross-dataset performance of the ImageNet-trained model poses a challenge to the existence of a relationship between the SEMAR score and generalization. This requires further investigation across different types of dataset.

4.5 Patch shuffling and model generalization

This experiment aims to investigate the impact of applying patch shuffling to the input images on model generalization. The same approach is used as in the experiment on model generalization (see Section 4.4). The only difference is an additional preprocessing step in which each image is first divided into patches, which are then randomly shuffled and recombined into a single image. This step is applied to the training, test and evaluation sets. The preprocessing technique aims to disrupt the semantic content of the images, possibly resulting in less overfitting on the semantic artifacts and better generalization.

4.5.1 Results

Table 3 presents the results of the patch shuffling experiment. It displays the performance of each model on all datasets in terms of both AP and accuracy. Moreover, each evaluation was performed three times.

The difference in performance between each run in both AP and accuracy did not exceed 0.5.

	CelebA		FFHQ		LSUN-Bed		LSUN-Church		ImageNet		Average	
Dataset	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.
CelebA	99.84	94.15	78.32	55.24	54.58	50.91	57.18	54.03	41.25	41.33	66.23	59.13
FFHQ	78.49	54.54	99.37	70.26	50.58	50.91	47.09	50.20	57.43	53.43	66.59	55.87
LSUN-Bed	58.1	52.72	78.93	77.12	98.52	90.42	62.16	50.20	42.88	46.88	68.12	63.47
LSUN-Church	75.83	68.25	51.15	50.60	47.07	48.08	98.67	92.14	50.45	49.40	64.63	61.69
ImageNet	47.85	50.40	54.00	50.30	42.95	50.20	51.75	50.60	96.74	89.01	58.66	58.10
Average	72.02	64.01	72.35	60.70	58.74	58.10	63.37	59.43	57.75	56.01	–	–

Table 3: Cross-dataset performance of each of the five datasets, using patch shuffling of the input data. Each row indicates a model trained on a dataset. The columns indicate the different test performances on the datasets.

4.5.2 Discussion

In order to establish a conclusion on the impact of patch shuffling on model generalization and the breaking of semantic artifacts, differences in model performances are examined between Table 2 and Table 3. One key observation is that patch shuffling appears to slightly decrease overall performance in both AP and accuracy. While some models, like the ones trained on ImageNet and LSUN-Bedroom, show a slight increase in overall performance, performance of models trained on the other datasets remains mostly the same or slightly decreases. Moreover, the diagonal performance of each model on a hold-out test set of the same dataset is overall slightly lower when applying patch shuffling. These observations could support the argument that by applying patch shuffling to the input images, classification models tend to overfit less to the dataset-specific biases. This, in turn, could lead to a more balanced generalization with lower performance in some areas such as on the hold-out test set of the same dataset, but slightly higher performance in other cross-domain tests.

The study by Zheng et al. [16] presents an experiment that examines the effects of patch shuffling on generalization, using the LSUN-Bedroom and LSUN-Church datasets. In their experiment, patch shuffling caused a significant increase in generalization for both models compared to models that did not use patch shuffling. Moreover, the experiment by Zheng et al. showed near-perfect performance when evaluated without patch shuffling. Additionally, their study used a ResNet50 model, whereas this experiment uses a less powerful ResNet18 model. These aspects could indicate that the model in their study relies more on the semantic artifacts of the training data, which may explain the larger increase in performance when using patch shuffling.

Furthermore, what stands out is the drop in accuracy of the model trained on FFHQ when applying patch shuffling. This drop in performance is mainly visible in the performance of the model on itself and on CelebA. This may indicate that the model trained on FFHQ relies on a type of bias or artifact that appears in both the CelebA and FFHQ datasets. By reducing the bias through patch shuffling, it is more difficult for the model to classify images of these datasets.

In contrast to the performance drop of the FFHQ model, the model trained on LSUN-Bedroom shows a significant increase in performance after applying patch shuffling. It shows an increase in

performance on all datasets, except on ImageNet. A possible explanation for this could be that the LSUN-Bedroom dataset contains more detailed and fine-grained structures in the images than FFHQ. By patch shuffling the images, the global structure that is distinctive for FFHQ and that this model relies on is disrupted. The local cues in LSUN-Bedroom, on the other hand, remain. This could make it easier for LSUN-Bedroom to generalize to other datasets, especially compared to FFHQ.

Additionally, the performance of each model in Table 3 and Table 2, with and without patch shuffling, and the SEMAR scores in Table 1 can be compared. This comparison can provide an indication of the relationship between the SEMAR score of a dataset and the effect patch shuffling has on the generalization of a model trained on that dataset. It can be observed that the LSUN-Bedroom dataset has a relatively high SEMAR score and shows a significantly large increase in generalization when applying patch shuffling. Moreover, the FFHQ dataset, with one of the relatively low SEMAR scores, shows a significant decrease in generalization. While these results could indicate a relationship between the SEMAR score and the effects of patch shuffling, the limited number of data points prevents drawing a reliable conclusion. Moreover, the LSUN-Church dataset exhibits the highest SEMAR score of all datasets used, but does not show a significant increase in generalization after applying patch shuffling. This also indicates the need for further investigation in future research.

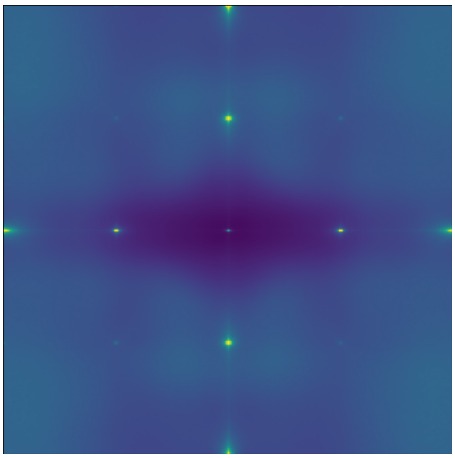
To conclude, applying patch shuffling seems to disrupt dataset-specific semantic artifacts. This is reflected in a decrease in in-domain performance, which could be a clear indication that the model is overfitting less on the dataset-specific biases. The effects on model generalization seem to differ per type of dataset. For a model trained on a dataset consisting of images with more fine-grained structure, like LSUN-Bedroom, it leads to better generalization. For models trained on other datasets, such as FFHQ, it decreases generalization performance. This could indicate that the effects of patch shuffling depend on whether the classification relies more on global or on local cues in the input images. Lastly, some of the results indicate the existence of a relationship between the SEMAR score of a dataset and the effects of patch shuffling on generalization. However, further experimentation on more diverse data is required in order to establish a reliable conclusion.

4.6 JPEG compression and model generalization

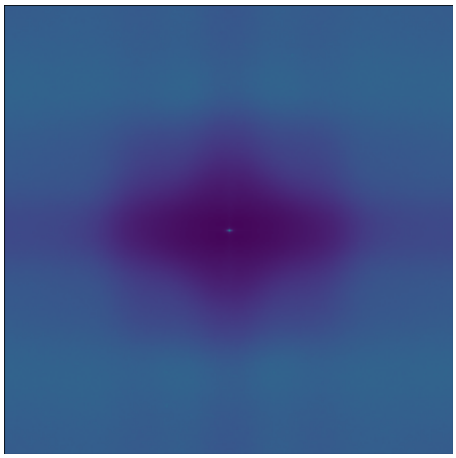
This additional experiment aims to demonstrate the effects of applying different preprocessing steps to the training data on the level of semantic artifacts. Moreover, their impact on model performance and generalization is examined. Specifically, the experiment examines the different effects of using PNG compression and JPEG compression during preprocessing. This preprocessing step is chosen for investigation, as JPEG compression is expected to be one of the preprocessing steps that most strongly influences semantic artifact scores.

4.6.1 Results

Figure 8 shows a comparison of using PNG and JPEG compression on the semantic artifacts in the FFHQ dataset. One spectrum is created from FFHQ with a JPEG compression preprocessing step, the other is created from FFHQ with a PNG compression preprocessing step. Moreover, Table 4 shows the performance of two models trained on these respective datasets, one trained on the JPEG-compressed version and one trained on the PNG-compressed version. This experiment is conducted using the same datasets and method as in Experiment 4.4.



(a) FFHQ, non-generated dataset with JPEG compression.



(b) FFHQ, non-generated dataset with PNG compression.

Figure 8: A comparison of spectra from two different variations of the real half of the FFHQ dataset, using PNG and JPEG compression.

	CelebA		FFHQ-JPEG		FFHQ-PNG		LSUN-Bed		LSUN-Church		ImageNet		Average	
Dataset	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.
FFHQ-JPEG	75.67	51.51	99.86	92.44	–	–	58.38	51.71	46.72	50.50	53.65	51.61	66.86	59.55
FFHQ-PNG	85.60	75.30	–	–	99.42	74.80	56.29	54.33	47.62	50.00	48.96	50.81	67.58	61.05

Table 4: Cross-dataset performance for FFHQ dataset using JPEG and PNG compression.

Dataset	SEMAR
FFHQ-JPEG	0.1574
FFHQ-PNG	0.1572

Table 5: SEMAR scores for the FFHQ dataset using JPEG and PNG compression

4.6.2 Discussion

The spectrum of the dataset that uses JPEG compression clearly shows an artifact in the form of a cross through the center. This artifact does not appear in the spectrum of the dataset that uses PNG compression. The influence on the SEMAR score, however, is minimal with a difference of only 0.0002 as can be seen in Table 5. Moreover, when looking at the performance of each model on the same dataset in Table 4, the FFHQ-JPEG model shows a significantly higher accuracy. This could support the idea that JPEG compression induces additional artifacts to the data, which makes it easier for a model to correctly classify images as real or fake.

Furthermore, the model trained on JPEG-compressed data shows lower average performance when tested on all datasets. This could also support the idea that JPEG compression leads to more overfitting on the training data and lower generalization. This is especially visible when comparing the performance of both models on the CelebA dataset. The notably higher performance of the FFHQ-PNG model could again suggest that training on PNG-compressed data encourages the model to learn more general features instead of overfitting on the compression artifacts, which ultimately leads to better generalization.

Based on these results, it can be concluded that JPEG compression has a measurable influence on model generalization and semantic artifacts. Future work could focus on further investigating this topic.

5 Limitations

5.1 Reliability of the SEMAR score

A key assumption related to the SEMAR score is that, by measuring the difference in angular variance between the noise residual power spectra of a dataset and a patch-shuffled version of that dataset, a metric can be obtained that correlates directly to the level of semantic artifacts in a dataset. Moreover, this assumes that applying patch shuffling to an image breaks the semantic structure of an image and that this effect is visible in the frequency power spectrum. This assumption is validated to some extent in Section 4.3. Overall, it has been observed that the computed SEMAR scores of the datasets reflect the relative ordering of the expected semantic artifact levels in the datasets based on their characteristics. Furthermore, it has been shown in Section 3.2.2 that patch shuffling has an effect on the intensity of the area of the frequency power spectrum that relates to the semantic artifacts in a dataset. However, this does not imply that this method is the optimal way of accurately describing semantic artifacts in a dataset. Future research could focus on further validating the current method, for example by evaluating it on different datasets.

Other limitations of the method for computing the SEMAR score include the choices for specific parameters. First, the difference in angular variance between the normal spectrum and shuffled spectrum is calculated only for the area that falls within a radius of 60 pixels from the center of the spectrum. Although this area is expected to cover most of the semantic artifacts in the spectrum, it is still arbitrary. Furthermore, the patch size of 32 x 32 pixels, which is used for shuffling the images, is selected based on outcomes of experiments in the research by Zheng et al. [16]. Although this parameter achieves optimal results in their study in terms of improving generalization and breaking semantic artifacts, it does not mean this transfers to the experiments in this study. Future research could include tests with several variations of these parameters to establish an optimal setup.

A further limitation concerns the generalization of the SEMAR score across different types of datasets and generative models. Although the current method shows a negative correlation between the SEMAR score of a training dataset and generalization of the classifier, it is unclear whether this trend holds for different types of datasets and generative models. In this study, only LDMs are used for generating the training data. However, it is unclear whether the SEMAR score still suffices as a reliable metric when applied to data generated by a GAN or VAE. Furthermore, it is unclear whether the method achieves similar performance when applied to datasets with levels of semantic variation outside the range of the five datasets used in this study. Applying the method in this study to the aforementioned types of datasets could further validate the SEMAR metric.

Lastly, no clear ground truth for the level of semantic artifacts in a dataset exists. Therefore, whether the SEMAR metric accurately captures the semantic artifacts in a dataset can only be partially validated. This was done by comparing SEMAR values of the datasets with an expectation of the levels of semantic artifacts, based on the characteristics of the dataset and outcomes of previous studies.

5.1.1 The influence of data preprocessing on Semantic Artifacts

Although the influence of preprocessing steps on semantic artifacts has been discussed in Sections 4.3 and 4.6, one aspect is still worth mentioning. Since every dataset used in the experiments contains the same preprocessing steps, including, for example, identical JPEG compression, it can be assumed that the

differences in SEMAR scores between the datasets are not influenced by preprocessing. However, there are certain aspects of the datasets used in this research that cannot be fully controlled. This includes the preprocessing of the data on which the generative models are trained, both for the LDMs and the autoencoder models. This also includes preprocessing steps performed on the real class of each dataset before they were collected for this research. Different preprocessing steps in both of these cases could potentially influence the artifacts of the dataset, the frequency power spectrum and consequently the SEMAR score. Therefore, differences in preprocessing steps between the datasets and their influence on SEMAR scores and possibly model generalization cannot be fully excluded in this research. Future work could involve training a generative model from scratch, in order to have full control of the training procedure and the types of artifacts that will be part of the output of the generative model.

Additionally, aspects of the datasets in this research such as image complexity and intra-class variation may influence both the SEMAR score and model generalization. For example, although the LSUN-Bedroom dataset contains only images of one specific class, images can vary a lot within this class. This could affect both generalization and the SEMAR score of that dataset. Therefore, the observed relationships between expected semantic artifacts in a dataset and its SEMAR score and generalization should be interpreted with caution.

6 Conclusion

This study investigated semantic artifacts, a specific type of bias in a dataset that can have an effect on the generalization of AI-generated image detection models. The aim of this study was to combine an analysis of the results of prior work with additional experiments, to gain a better understanding of semantic artifacts and their effects on generalization of AI-generated image classifiers. The study consisted of three main experiments, each of which aimed to answer one of the research questions. The key findings and conclusions for each research question are stated below:

RQ1: How can semantic artifacts in a dataset be described in a quantitative way? This study established a method to describe the semantic artifacts in a dataset via a single metric: the SEMAR score. This metric is based on the analysis of frequency power spectra of the datasets which, in this study, have been shown to accurately reflect the specific characteristics and biases of the datasets from which they are derived (Section 4.3). Furthermore, it has been demonstrated that applying patch shuffling to the images of the dataset partially reduces the intensity of the semantic artifacts visible in the frequency power spectrum. Since no ground truth for the level of semantic artifacts in a dataset exists, the SEMAR score can only be validated by a comparison with expected results. Nevertheless, the results in this study are promising for reliably capturing the characteristics and semantic bias of a dataset in a single metric.

RQ2: How do semantic artifacts in the training data affect a model's ability to generalize? The results of the experiments conducted in this research indicate a slight negative correlation between the level of semantic artifacts in a training dataset and the ability of a classifier model to generalize well to other types of datasets (Section 4.4). However, given the limited number of data points, no strong conclusions can be drawn from this result. Furthermore, the observed generalization gap increases as the images of the training dataset and the evaluation dataset differ more in terms of semantic content such as the overall scene, composition or type of object. The results indicate that classifiers tend to generally perform near perfectly when evaluated on a dataset of the same distribution, and that this is independent of the level of semantic artifacts of the dataset.

RQ3: What is the impact of applying patch shuffling to a dataset on its level of semantic artifacts and the generalization of a model trained on it? Through the experiments conducted in this research, it has been demonstrated that applying patch shuffling to the images in a dataset disrupts the dataset-specific semantic artifacts in it (Section 4.5). This is reflected in two ways: First, the in-domain performance of classifier models is generally slightly decreased when applying patch shuffling, indicating that classifiers are generally overfitting less on the dataset-specific biases. Second, for several models, generalization is increased after applying patch shuffling. This is especially the case for models that are trained on datasets with fine-grained structure and a high level of detail, such as a dataset consisting of images of bedrooms. This indicates that the effects of patch shuffling depend on the type of dataset and whether the classification model relies more on global or on local cues in the input images.

Together, these findings provide a better understanding of how semantic artifacts and dataset bias affect AI-generated image detection. Moreover, they provide a clear and concrete method for measuring and mitigating the semantic artifacts in a training dataset for AI-generated image detection models. Future work could involve further evaluation and improvement of the methods used (Section 5), for example by testing it on datasets generated by different types of models and on datasets with greater semantic variation. Additionally, the effects of including different preprocessing steps, such as JPEG compression, augmentation and random cropping, on semantic artifacts and model generalization could be investigated.

References

[1] A. Singla, A. Sukharevsky, B. Hall, L. Yee, M. Chui, and T. Balakrhisnan, "The state of AI in 2025: Agents, innovation, and transformation," tech. rep., McKinsey & Company, November 2025. McKinsey Global Survey.

[2] Stanford HAI, "The 2025 AI index report," tech. rep., Stanford University Human-Centered Artificial Intelligence, 2025. AI Index Report.

[3] S. Sharma, "Benefits or concerns of ai: A multistakeholder responsibility," *Futures*, vol. 157, p. 103328, 2024.

[4] M. Brundage, S. Avin, J. Clark, H. Toner, P. Eckersley, B. Garfinkel, A. Dafoe, P. Scharre, T. Zeitzoff, B. Filar, *et al.*, "The malicious use of artificial intelligence: Forecasting, prevention, and mitigation," *arXiv preprint arXiv:1802.07228*, 2018.

[5] E. Altuncu, V. N. Franqueira, and S. Li, "Deepfake: definitions, performance metrics and standards, datasets, and a meta-review," *Frontiers in Big Data*, vol. 7, p. 1400024, 2024.

[6] L. Verdoliva, "Media forensics and deepfakes: an overview," *IEEE journal of selected topics in signal processing*, vol. 14, no. 5, pp. 910–932, 2020.

[7] X. Zhang, S. Karaman, and S.-F. Chang, "Detecting and simulating artifacts in gan fake images," in *2019 IEEE international workshop on information forensics and security (WIFS)*, pp. 1–6, IEEE, 2019.

[8] J. Ricker, S. Damm, T. Holz, and A. Fischer, "Towards the detection of diffusion model deepfakes," *arXiv preprint arXiv:2210.14571*, 2022.

[9] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, "Cnn-generated images are surprisingly easy to spot... for now," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8695–8704, 2020.

[10] S. Yan, O. Li, J. Cai, Y. Hao, X. Jiang, Y. Hu, and W. Xie, "A sanity check for ai-generated image detection," *arXiv preprint arXiv:2406.19435*, 2024.

[11] R. Corvi, D. Cozzolino, G. Zingarini, G. Poggi, K. Nagano, and L. Verdoliva, "On the detection of synthetic images generated by diffusion models," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, IEEE, 2023.

[12] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

[13] U. Ojha, Y. Li, and Y. J. Lee, "Towards universal fake image detectors that generalize across generative models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24480–24489, 2023.

[14] F. Guillaro, G. Zingarini, B. Usman, A. Sud, D. Cozzolino, and L. Verdoliva, "A bias-free training paradigm for more general ai-generated image detection," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18685–18694, 2025.

[15] Z. Zhou, Y. Luo, Y. Wu, K. Sun, J. Ji, K. Yan, S. Ding, X. Sun, Y. Wu, and R. Ji, "Aigi-holmes: Towards explainable and generalizable ai-generated image detection via multimodal large language models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18746–18758, 2025.

[16] C. Zheng, C. Lin, Z. Zhao, H. Wang, X. Guo, S. Liu, and C. Shen, "Breaking semantic artifacts for generalized ai-generated image detection," *Advances in Neural Information Processing Systems*, vol. 37, pp. 59570–59596, 2024.

[17] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[18] I. J. 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.

[19] A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, *et al.*, "Conditional image generation with pixelcnn decoders," *Advances in neural information processing systems*, vol. 29, 2016.

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

[21] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz, "Leveraging frequency analysis for deep fake image recognition," in *International conference on machine learning*, pp. 3247–3258, PMLR, 2020.

[22] H. Zhao, W. Zhou, D. Chen, T. Wei, W. Zhang, and N. Yu, "Multi-attentional deepfake detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2185–2194, 2021.

[23] L. Chai, D. Bau, S.-N. Lim, and P. Isola, "What makes fake images detectable? understanding properties that generalize," in *European conference on computer vision*, pp. 103–120, Springer, 2020.

[24] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *CVPR 2011*, pp. 1521–1528, IEEE, 2011.

[25] Z. Liu and K. He, "A decade's battle on dataset bias: Are we there yet?," *arXiv preprint arXiv:2403.08632*, 2024.

[26] Z. Yang, R. Chen, Z. Yan, K.-Y. Zhang, X. Fu, S. Wu, X. Shu, T. Yao, S. Ding, and X. Li, "All patches matter, more patches better: Enhance ai-generated image detection via panoptic patch learning," *arXiv preprint arXiv:2504.01396*, 2025.

[27] S. Gye, J. Ko, H. Shon, M. Kwon, and J. Kim, "Sfld: Reducing the content bias for ai-generated image detection," *arXiv preprint arXiv:2502.17105*, 2025.

[28] B. Chu, W. You, M. Li, T. Zheng, K. Zhao, X. Xu, Z. Lu, J. Song, M. Xu, and L. Zhou, "When semantics regulate: Rethinking patch shuffle and internal bias for generated image detection with clip," *arXiv preprint arXiv:2511.19126*, 2025.

[29] N. Zhong, Y. Xu, S. Li, Z. Qian, and X. Zhang, "Patchcraft: Exploring texture patch for efficient ai-generated image detection," *arXiv preprint arXiv:2311.12397*, 2023.

[30] R. Corvi, D. Cozzolino, G. Poggi, K. Nagano, and L. Verdoliva, "Intriguing properties of synthetic images: from generative adversarial networks to diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 973–982, 2023.

[31] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz, "Leveraging frequency analysis for deep fake image recognition," in *International conference on machine learning*, pp. 3247–3258, PMLR, 2020.

[32] A. Torralba, P. Isola, and W. Freeman, *Foundations of Computer Vision*. Adaptive Computation and Machine Learning series, MIT Press, 2024.

[33] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[34] M.-A. Parseval, "Mémoire sur les séries et sur l'intégration complète d'une équation aux différences partielles linéaires du second ordre, à coefficients constants," *Mém. prés. par divers savants, Acad. des Sciences, Paris,(1)*, vol. 1, no. 638-648, p. 42, 1806.

[35] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE transactions on image processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[36] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.

[37] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[38] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[39] 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.

[40] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv preprint arXiv:1506.03365*, 2015.

[41] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, *et al.*, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *International journal of computer vision*, vol. 128, no. 7, pp. 1956–1981, 2020.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

A Appendix

A.1 Dataset details

Property	CelebA	FFHQ	LSUN-Church	LSUN-Bedroom	ImageNet
No. images	10,000	10,000	10,000	10,000	10,000
No. classes	1	1	1	1	1,000
Subject	Human faces	Human faces	Churches	Bedrooms	General objects
Scene type	Portrait	Portrait	Outdoor	Indoor	General
Object scale	Close-up	Close-up	Mid–wide	Mid–wide	Varies
Camera angle	Frontal	Frontal	Varies	Varies	Varies
Background variability	Low	Medium-Low	Medium	Medium	High
Homogeneity	High	Medium-High	Medium	Medium	Low

Table 6: Characteristics of the datasets used in this research, based on metadata and observations.

A.2 Dataset generation details

Table 7 shows an overview of the different model types and hyperparameters that are used for generating the datasets in this research.

Dataset	η	DDIM Steps	Type	Autoencoder
CelebA	0	200	Unconditional	VQ-4
FFHQ	0	100	Unconditional	VQ-4
LSUN-Church	0	250	Unconditional	KL-8
LSUN-Bedroom	0	200	Unconditional	VQ-4
ImageNet	0	100	Class-conditional	VQ-8

Table 7: Model types and hyperparameters used for image generation per dataset.