Diffusion-Based Inpainting of Corrupted Spectrogram

MAHSA MASSOUD

School of Computer Science

McGill University, Montreal

December, 2024

A thesis submitted to McGill University in partial fulfillment of the

requirements of the degree of Master of Computer Science

Abstract

A spectrogram is a visual representation of the spectrum of frequencies in a signal as it varies with time. A critical problem in radio astronomy is removing radio frequency interference (RFI) from the spectrograms produced by radio telescopes. Given the performance of diffusion models in image inpainting and the structural similarity of spectrograms to image data, these methods can be useful for addressing RFI corruption. However, applying diffusion models to this problem presents significant challenges. In particular, the astronomical data we are dealing with is corrupted, which makes the existing methods that depend on clean training data inapplicable.

This thesis explores methods to improve solutions to the problem of spectrogram inpainting, specifically designed for scenarios where all training data is corrupted. We further propose a positional encoding scheme to address the assumption of translation symmetry, in Convolution-based architectures such as UNET, and enable the models to capture frequency-dependent patterns in spectrograms better.

We evaluate our methods on CIFAR-10, synthetic spectrograms, and real-world spectrograms from radio astronomy. Our results demonstrate the effectiveness of these approaches in reconstructing corrupted data, highlighting the potential of diffusion-based inpainting for spectrograms. This work provides a foundation for applying generative models in astrophysical data recovery and paves the way for further exploration in this domain.

Sommaire

Un spectrogramme est une représentation visuelle du spectre des fréquences d'un signal au fil du temps. Un problème majeur en astronomie radio est l'élimination des interférences radiofréquences (RFI) dans les spectrogrammes produits par les radiotélescopes. Étant donné les performances des modèles de diffusion pour le inpainting d'images et la similarité structurelle entre les spectrogrammes et les données d'image, ces méthodes peuvent être utiles pour traiter la corruption causée par les RFI. Cependant, appliquer ces modèles de diffusion à ce problème présente des défis importants. En particulier, les données astronomiques avec lesquelles nous travaillons sont corrompues, ce qui rend les méthodes existantes dépendant de données d'entraînement propres inapplicables.

Cette thèse explore des méthodes visant à améliorer les solutions pour le inpainting de spectrogrammes, spécifiquement conçues pour des scénarios où l'ensemble des données d'entraînement est corrompu. Nous proposons également un schéma d'encodage positionnel pour répondre à l'hypothèse de symétrie par translation et permettre aux modèles de mieux capturer les motifs dépendant des fréquences dans les spectrogrammes.

Nous évaluons nos méthodes sur les ensembles de données CIFAR-10, des spectrogrammes synthétiques et des spectrogrammes réels issus de l'astronomie radio. Nos résultats démontrent l'efficacité de ces approches pour reconstruire des données corrompues, soulignant le potentiel des techniques de inpainting basées sur les modèles de diffusion pour les spectrogrammes. Ce travail pose les bases de

l'application de modèles génératifs pour la récupération de données astrophysiques et ouvre la voie à des explorations futures dans ce domain.

Previously Published Material

This thesis is based on work previously published in the Machine Learning for the Physical Sciences (ML4PS) Workshop at NeurIPS 2024, titled *"Diffusion-Based Inpainting of Corrupted Spectrogram"*. The core ideas and methods presented in this thesis, particularly those in Chapters 3, originated from this publication.

I was the primary contributor to the work, leading the research from problem formulation, algorithm development, and experimentation on astronomical spectrograms to the writing of the paper. My co-authors supported the research through valuable discussions and feedback. This thesis extends the workshop paper with a more thorough literature review, enhanced methodology, and additional experimental analysis.

Dedicated to my beloved parents and sister,

whose unwavering love, and support carried me through every step of this journey.

And to my wonderful friends,

for their encouragement, laughter, and companionship.

Acknowledgments

To begin, I want to sincerely thank my supervisor, Prof. Siamak Ravanbakhsh, for his unwavering support, mentorship, and flexibility. I am deeply grateful for the opportunity to be part of his lab at McGill University and the Mila community. His guidance and encouragement have been invaluable in helping me build the best path for my career and fostering my growth in a research environment. Under his guidance, I have learned not only about research but also about making thoughtful life decisions.

Special thanks to Prof. Adrian Liu, whose guidance on the interdisciplinary aspects of this project was inspiring. I also want to extend my gratitude to my collaborators, especially Reyhane Askari, for her amazing guidance, kind heart, and encouragement. Her support and positivity have made this journey even more meaningful.

I also want to thank the amazing community of the McGill Computer Science Graduate Society (CSGS) and the Mila-Quebec AI Institute. Both became like a second home where I built lasting friendships, made meaningful connections, picked up new hobbies, and shared incredible travel experiences—and so much more.

Finally, a special thank you goes to my family, who I really miss and wish could be here with me right now. Their unwavering support has been my strength. Finally, to my friends, both in Montreal and around the world, thank you for making these past two years so enjoyable and unforgettable. You've made this journey truly special.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

RFI	Radio Frequency Interference
DDPM	Denoising Diffusion Probabilistic Model
GAN	Generative Adversarial Network
VAE	Variational Autoencoder
UNet	U-shaped Convolutional Neural Network
MSE	Mean Squared Error
PSNR	Peak Signal-to-Noise Ratio
HERA	Hydrogen Epoch of Reionization Array
CNN	Convolutional Neural Network
DPSS	Discrete Prolate Spheroidal Sequences
GPR	Gaussian Process Regression
LSSA	Least Square Spectral Analysis
NLN	Nearest-Latent-Neighbours
PDE	Partial Differential Equation

Chapter 1

Introduction

1.1 Background and Motivation

1.1.1 Inpainting

Inpainting is a fundamental problem in computer vision and deep learning with widespread applications across various fields such as image restoration, medical imaging, and digital content creation. The primary goal of inpainting is to reconstruct missing or corrupted parts of data in a way that is coherent and consistent with the surrounding information. This task plays a crucial role in enhancing the quality of images and improving the accuracy of subsequent analyses in many domains.

In computer vision research, the quality and completeness of images significantly impact some important downstream tasks such as object detection, segmentation, and scene understanding. Missing data or corrupted regions in images can lead to inaccurate results in these tasks, potentially compromising the reliability of AI systems. Inpainting techniques help to restore these missing parts, ensuring that the data remains useful for further analysis and processing. The ability to effectively reconstruct missing information has far-reaching implications, from

improving the robustness of vision systems to enabling new applications, image editing, image interpretation ability, etc.

A novel approach to tackle the inpainting problem is to train diffusion models exclusively on corrupted data to generate reconstructed, filled-in images. This method represents a significant departure from previous techniques that rely on paired datasets of corrupted and original images. The core idea is to develop a model capable of learning the underlying structure and patterns of images from corrupted data alone. By doing so, the model can potentially generalize better to real-world scenarios where the original, uncorrupted data may not be available. This approach is particularly relevant in fields such as astronomy, where certain data may be inherently corrupted or incomplete due to various factors like Radio Frequency Interference (RFI). The training process involves feeding the model with corrupted images and optimizing it to predict the noise added during the forward diffusion process. Through iterative refinement in the backward process, the model learns to denoise and reconstruct the image, effectively filling in missing or corrupted regions. This thesis aims to explore and develop such a model, leveraging the power of diffusion-based approaches while addressing the unique challenges posed by training solely on corrupted data. The potential benefits of this method include:

- Improved generalization to real-world, naturally corrupted data specifically in the astronomical signals
- Potential for application in domains where uncorrupted data is scarce or unavailable

By focusing on this innovative approach, we aim to contribute to the field of image inpainting and explore its applications in areas such as astronomical data reconstruction.

Before the rise of deep learning, traditional inpainting methods were widely used. These methods can be broadly categorized into techniques based on Partial Differential Equations (PDEs), patch-based methods, and graphical models. Techniques based on PDEs , as demonstrated by Bertalmio et al [2], propagate information from the edges and boundaries to achieve smooth inpainted images. Patch-based methods, such as those proposed by [9], try to find and replicate similar patches from known areas to reconstruct the missing regions. Graphical models, including Markov Random Fields and Conditional Random Fields, have also been extensively used to represent and infer missing parts based on probabilistic dependencies within the data [12, 35].

Deep learning has revolutionized the field of image inpainting, enabling the development of models that can handle more complex scenarios with higher accuracy. These techniques can be categorized based on the underlying architectures and methodologies. Convolutional Neural Networks (CNNs) are a class of feed-forward neural networks that have been extensively used for image inpainting due to their ability to capture spatial hierarchies in images.

In astronomical research, the quality of celestial images significantly impacts the analysis and measurement of features. One major challenge is Radio Frequency Interference (RFI), which necessitates frequent data flagging in radio photon measurements. RFI poses significant challenges for current and future radio telescopes, with the number, variety, and overall disruption to observations increasing [8]. These interfering signals, primarily originating from terrestrial transmitters and satellites, often result in certain data being marked as potentially important during analysis. RFI can corrupt weak cosmic signals and severely impact the quality of astronomical data [28]. The complex nature of RFI, as summarized by [28], includes:

- Increasing prevalence due to the growing number of electronic devices

- Varied sources, including internal (generated by instruments) and external (man-made radio emissions)

Removing RFI from data analysis is crucial, but it introduces gaps that can cause artifacts within the spectrum. These gaps pose a direct challenge to data analysis pipelines that strive to separate foregrounds from cosmological signals in the Fourier domain. This scenario can be framed as an inpainting problem in computer vision, aiming to fill these gaps coherently with the rest of the observed data.

Application to Astrophysical Data

In the context of this thesis, we focus on the application of inpainting techniques to astrophysical data, specifically from the Hydrogen Epoch of Reionization Array (HERA) experiment. HERA is a large radio interferometer consisting of 300 radio dishes, located in the South African Karoo Desert. Its goal is to characterize the three-dimensional spatial properties of our Universe around the time that first-generation galaxies were forming. The raw data from HERA comes in the form of complex-valued visibilities, which are functions of frequency and time. These visibilities can be viewed as two-dimensional functions on the frequency-time plane, analogous to images. However, they are often contaminated by Radio Frequency Interference (RFI), which is orders of magnitude brighter than any astrophysical source of radio waves. This contamination leads to parts of the data being masked, creating a scenario similar to traditional image inpainting problems.

Unique Challenges in Astrophysical Data Inpainting

Two major challenges distinguish this problem from general inpainting tasks: The entire training dataset is corrupted by RFI, necessitating novel approaches to learn the true signal distribution. Unlike typical image data, spectrograms lack translation symmetry along the frequency axis, which is inconsistent with the assumptions of convolutional neural networks. To address these challenges, we propose

three progressively more accurate solutions and introduce a positional encoding scheme for frequencies. This approach aims to improve the model's ability to handle the unique characteristics of astronomical spectrograms while maintaining the benefits of convolutional architectures.

1.1.2 Radio Frequency Interference (RFI)

Radio Frequency Interference (RFI) is a significant challenge in radio astronomy data collection. It refers to any unwanted radio signals that interfere with the weak cosmic signals astronomers want to study. RFI can originate from various sources, both internal (generated by instruments) and external (man-made radio emissions) [28].

The impact of RFI on astronomical observations is substantial. It can corrupt weak cosmic signals, potentially impacting the integrity and scientific value of the collected data [28]. As radio telescopes become more sensitive, the problem of RFI is expected to become even more pronounced [28].

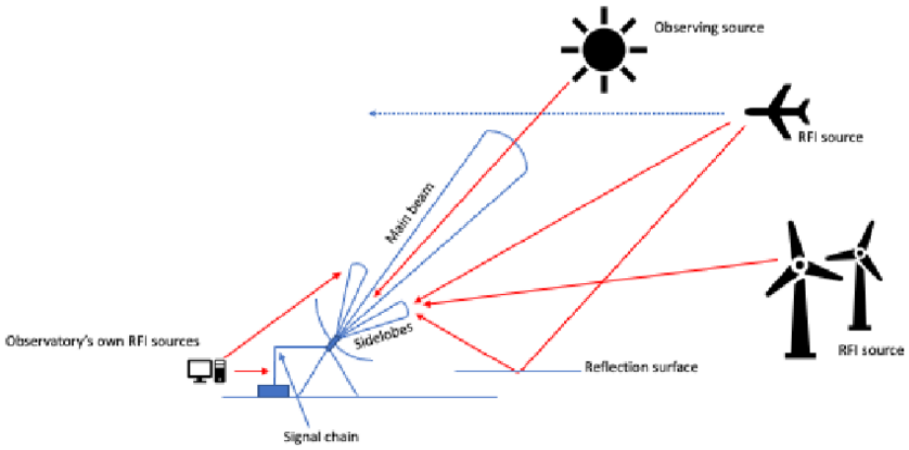

Figure 1.1: RFI from different sources [31]

RFI can manifest in different forms:

1. Narrow band interference: This includes continuous wave (CW) or modulated CW signals, often appearing as narrow vertical lines or slightly wider modulated vertical bands on a spectrum analyzer [39].

2. Broadband interference: This typically includes emissions from switch-mode power supplies, such as electrical discharges from power cables, or digital systems like Wi-Fi or Bluetooth. On a spectrum analyzer, it appears as broad ranges of signals or an increase in the noise floor [39].

The detection and mitigation of RFI have become crucial tasks in radio astronomy. Traditional methods often involve manual inspection and flagging of contaminated data, but this approach becomes impractical with the increasing volume of data from modern radio telescopes. As a result, there's a growing interest in automated RFI detection methods, particularly those leveraging machine learning and deep learning techniques [28].

Recent advancements in this field include the application of convolutional neural networks, generative adversarial networks, and other sophisticated algorithms to identify and characterize RFI more effectively [28]. These approaches aim to improve the accuracy and efficiency of RFI detection, ultimately enhancing the quality and reliability of radio astronomical observations.

1.2 Problem Statement

To address the effects of Radio Frequency Interference (RFI) on the power spectrum, a common approach in radio astronomy is to detect RFI-affected frequency bands and then mitigate its impact by avoiding those bands entirely. Detection is about identifying where the interference occurrence is, and mitigation includes techniques used to reduce or eliminate its impact on the data. While this approach works, it often comes at the cost of losing valuable frequency channels, which limits its analysis and reduces the quality of the results. Consequently, researchers face

restrictions in accessing different redshifts, which can lead to a reduced signal-to-noise ratio. This reduction makes data interpretation and analysis complicated. It can also cause challenges in extracting meaningful cosmological information from the observations.

Traditional RFI detection and mitigation techniques, such as CUMSUM [3], singular value decomposition [13], and wavelet-based methods, have been widely implemented in real-time at observatories [28]. These methods generate RFI masks for archived data, but they come with some limitations. They often lack adaptability to dynamic or evolving RFI patterns and can be sensitive to the choice of parameters. Furthermore, traditional techniques may struggle with complex RFI signals, particularly those that exhibit Gaussian or near-Gaussian distributions. The computational demands of processing large datasets in real-time also pose significant challenges [32].

In contrast, recent advancements in deep learning have shown promise in improving RFI detection. Supervised learning methods have demonstrated effectiveness; like how authors of [1] train U-Net architecture in a supervised manner; however, they require extensive labeled datasets, which can be costly and impractical to obtain. Additionally, these models may face issues with generalization, as they can overfit specific telescope data or frequency ranges, and their "black box" nature can hardly be interpretable [8].

Unsupervised learning approaches offer a potential alternative by reducing dependence on labeled data. For instance, methods like the Nearest-Latent-Neighbours (NLN) algorithm frame RFI detection as an anomaly detection task; this can let the learning benefit from the vast amount of existing radio telescope data [30]. However, these methods are not without their challenges. They may produce false positives by misidentifying unusual but valid astronomical signals as RFI, and their implementation can be complex.

In summary, while traditional methods and some deep learning approaches for RFI mitigation and detection have made progress, they still have limitations. This indicates a need for better techniques to handle RFI in astronomical data.

1.3 Objective of the Study

The objective of this study is to develop an effective inpainting method using Denoising Diffusion Probabilistic Models (DDPMs) to address the issues introduced by RFI in astronomical data. This research aims to design a proper setup that generates unmasked images without corruption from natural datasets by training diffusion models on corrupted data.

To achieve the best possible inpainting results, the study will focus on testing and optimizing three main steps in the proposed inpainting algorithm iteratively. This iterative process (Inpainting) will ensure that the inpainting method can effectively restore the integrity of the data while maintaining coherence with the existing imagery.

Finally, the effectiveness of the proposed method will be validated using actual astronomical datasets such as HERA. The goal is to demonstrate the potential of the inpainting approach to improve the quality of the astronomical dataset and improve the accuracy of astronomical analyses.

1.4 Contributions of the Thesis

This thesis makes the following contributions:

- A novel pipeline that employs diffusion models for inpainting corrupted astronomical data, trained on datasets with deliberately introduced masks.

- An iterative algorithm with three key steps that enhance the quality of inpainting results, ensuring seamless integration of new content with existing data.
- Experimental validation of the proposed method on astronomical datasets, highlighting its efficacy in restoring continuous and coherent images, thereby improving the quality of astronomical analyses and feature measurements.

Chapter 2

Literature Review

2.1 Inpainting Methods

Inpainting is the process of filling in missing or damaged parts of an image. There are several common approaches to this problem. Traditional methods include techniques based on mathematical equations, for example extending edges to fill gaps [2], and patch-based methods that copy similar textures or patterns [4]from surrounding areas. While these methods can work well in simple cases, they often fail when dealing with complex or irregular gaps. Deep learning has introduced more advanced techniques, such as generative models like GANs or autoencoders, which learn patterns from data to generate realistic reconstructions. Among these approaches, diffusion-based methods have become particularly effective. These methods use an iterative process to refine the missing regions step by step, achieving highly accurate results. Diffusion-based inpainting can be grouped into three main types: modifying the sampling process, modifying the training process, or combining both. Our work builds on this last category to address the specific challenges of reconstructing corrupted spectrograms.

2.1.1 Generative Models

Generative models are a class of machine learning models that aim to generate new data samples that resemble a given dataset. These models learn the distribution of the data and can produce new instances that are statistically similar to the training data. Generative models have numerous applications in various fields, including image generation, text synthesis, data augmentation, etc.

Types of Generative Models

There are several types of generative models, each with its unique approach to learning and generating data:

Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) [15] consist of two neural networks: a generator and a discriminator. The generator creates synthetic data samples, while the discriminator evaluates their authenticity compared to real data. The two networks are trained simultaneously in a competitive manner, where the generator aims to produce realistic samples to fool the discriminator, and the discriminator aims to distinguish between real and generated samples. GANs have been widely used for image generation, inpainting, and style transfer.

Variational Autoencoders (VAEs)

Variational Autoencoders (VAEs) [26] is a type of generative model that extends traditional autoencoders by introducing a probabilistic latent space. Instead of encoding the input into a fixed vector, VAEs encode the data as a distribution in the latent space, typically represented by a mean (μ) and variance (σ^2):

$$z \sim \mathcal{N}(\mu, \sigma^2), \tag{2.1}$$

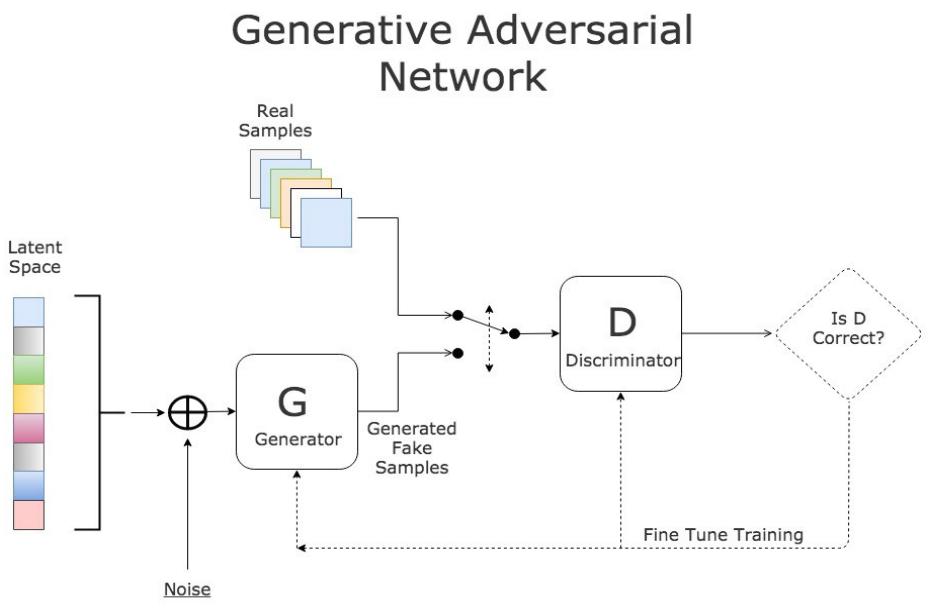

Figure 2.1: generative Adversarial Models [15]

where z is the latent variable sampled from the learned distribution. The decoder then reconstructs the input by sampling from this latent representation. To train a VAE, the loss function includes both a reconstruction loss, ensuring the output resembles the input, and a regularization term, the Kullback-Leibler (KL) divergence, to enforce the latent space to follow a standard normal distribution:

$$\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{reconstruction}} + \beta\, D_{\text{KL}}(q(z|x)\|p(z)). \tag{2.2}$$

Here, $q(z|x)$ is the learned posterior, $p(z)$ is the prior (typically $\mathcal{N}(0,1)$), and β balances the two terms. This probabilistic framework allows VAEs to generate new data by sampling from the latent space, distinguishing them from traditional autoencoders.

Denoising Diffusion Probabilistic Models (DDPMs)

Denoising Diffusion Probabilistic Models (DDPMs) [38] are a class of generative models that iteratively apply denoising steps to reduce noise in data samples. Starting from a noisy sample, DDPMs progressively refine the sample through a

series of denoising steps until a clean data sample is obtained. These models have shown remarkable performance in generating high-quality images and have been successfully applied to image inpainting tasks.

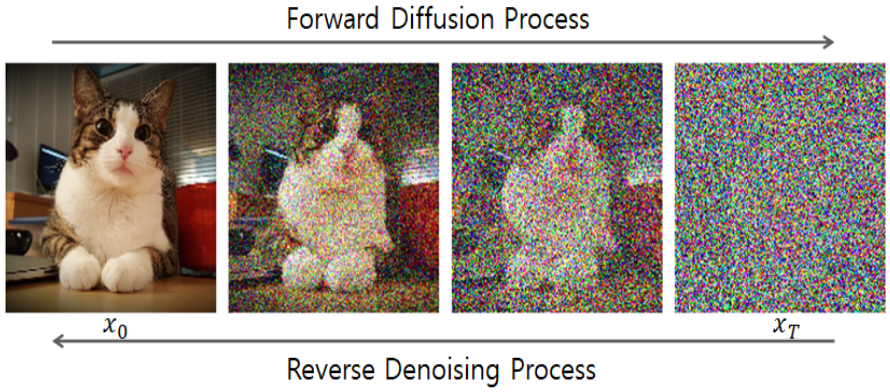

Figure 2.2: DDPM Pipeline [20]

2.1.2 Applications of Generative Models in Image Inpainting

Generative models have significantly advanced the field of image inpainting by providing robust methods to fill in missing or corrupted regions of images. The ability of these models to learn complex data distributions and generate realistic samples makes them ideal for inpainting applications.

GAN-based Inpainting

GAN-based inpainting methods leverage the adversarial training mechanism to generate realistic image completions. For example, the AOT-GAN model [45] enhances context reasoning and texture synthesis by aggregating contextual transformations from various receptive fields. This approach allows the model to capture both distant image contexts and fine-grained textures, resulting in high-quality inpainted images.

VAE-based Inpainting

Diffusion Model-based Inpainting

Diffusion model-based inpainting methods, such as the RePaint approach [29], iteratively denoise images to fill in missing regions. These methods leverage the powerful denoising capabilities of DDPMs to generate inpainted images with high accuracy and detail. The iterative nature of the process ensures that the inpainted regions align well with the known parts of the image.

On top of the above, we can categorize the inpainting method into three main methods of modification, based on training and sampling time.

2.1.3 Methods that Modify the Sampling

Sampling modifications involve altering the denoising steps in diffusion models to better align inpainted regions with the known unmasked areas. In [16], the authors introduce a method called GradPaint, which uses a custom loss to measure the coherence of the denoised image estimation with the masked input image at each step in the denoising process. This includes a mean squared error loss outside the inpainting mask and an alignment loss to maintain smooth transitions at the mask boundaries. Similarly, RePaint [29]conditions the diffusion process on known regions, iteratively improving the reconstruction of masked areas. This approach can be applied to any pre-trained diffusion model and only requires modifying the denoising scheduling of DDPM for inpainting. In another work, the authors of [44] suggest a different approach for sampling; they designed a multimodal inpainting framework that combines diffusion models with text and shape guidance. While it wasn't explicitly trained on every modality, its use of pre-trained diffusion models and flexible conditioning mechanisms allows it to adapt to various types of data, including text-guided and shape-guided inpainting tasks. This general-

ization capability makes it versatile across data types. However, Uni-Paint lacks frequency-specific adaptations, which limits its applicability to spectrograms.

2.1.4 Methods that Modify the Training

Training modifications focus on adjusting the learning process of diffusion models to ensure the reconstructed regions align seamlessly with the unmasked areas In [36], the authors guide the model to fill in masks based on the context provided by the existing parts of the image. This approach refines the training process by introducing conditional loss functions that guide the model to generate contextually consistent results. This allows the model to leverage the information in the visible regions to predict and generate the missing parts. Another example is LatentPaint [42] which modifies the latent space representation during training to improve reconstruction efficiency, especially for irregular gaps. However, spectrograms require handling unique frequency-specific while training the model.

2.1.5 Methods that Modify Both Sampling and Training

In [6], the authors propose a new approach for image inpainting using diffusion models that do not require expensive training and are fast at inference time. This is achieved by performing the forward-backward diffusion step in the latent space rather than the image space. A mask is applied to the latent space representation to simulate the missing parts. During training, a loss function measures the difference between the reconstructed latent space (after inpainting) and the original latent space. During inference, the trained diffusion model tries to generate the inpainted representation for the masked regions.

By going through recent works in the field of computer vision, we can observe that generative models have a significant impact by enabling the creation of highly realistic images and reconstructions. These models, including GANs, au-

toencoders, and diffusion-based frameworks, form the foundation for many modern inpainting techniques. Generative models have a great impact on the various domains of computer vision, enabling the creation of highly realistic images including inpainting methods. The next section delves into the principles and applications of generative models, highlighting their significance in the field of image inpainting.

2.1.6 Traditional methods for Inpainting

Before the era of deep learning, traditional inpainting methods laid the groundwork for filling missing regions in images. According to the survey by these techniques can be broadly categorized into diffusion-based methods, patch-based approaches, and exemplar-based methods, each with its unique advantages and limitations.

In the figure below, the diagram illustrates the summary of the methods:

Diffusion-Based Methods: These methods rely on propagating pixel information from known areas into missing regions, using mathematical principles such as Partial Differential Equations (PDEs). Early works, like those by Bertalmio et al. [5] and Ballester et al. [2], formulated inpainting as a smooth continuation of image structures, that were effective for smooth filling of images structures. these methods were basically used for repairing small gaps or extending simple edges. Tschumperlé [40] advanced these ideas by introducing anisotropic diffusion, which enhanced the stability and directional coherence of the reconstructed areas. In this diffusion process, the algorithm adaptively follows the local geometry of the image, to preserve sharp edges while smoothing out noise in homogeneous regions. A common PDE used in this context is:

$$\frac{\partial I}{\partial t} = \nabla \cdot (c(x, y)\nabla I), \tag{2.3}$$

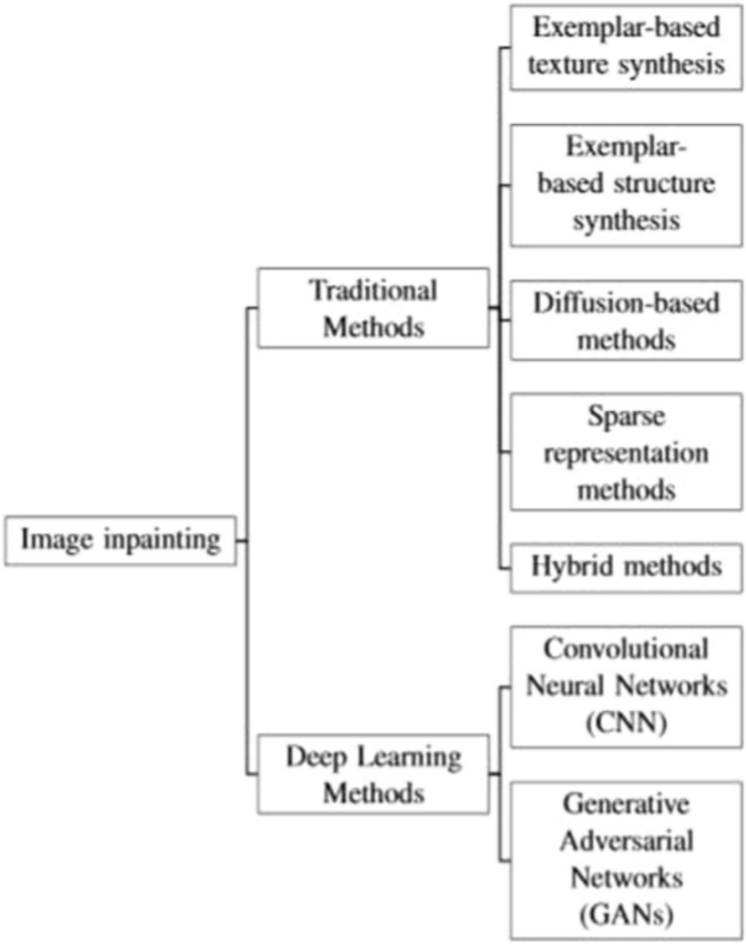

Figure 2.3: Image Inpainting Methods

Patch-Based Approaches: Patch-based methods reconstruct missing regions by copying patches from known areas with similar textures. Efros and Leung [9] utilized this idea of to create smooth textures. Building on this, Barnes et al. [4] introduced PatchMatch, which accelerated the search for similar patches. Later, methods like those by Darabi et al. [7] and Huang et al. [22] improved patch blending, allowing for better reconstructions of highly textured or complex regions.

If we show known parts with (P_k) and unknown regions with(P_u):

$$E(P_k, P_u) = \sum_{i,j} \|P_k(i,j) - P_u(i,j)\|^2, \tag{2.4}$$

Exemplar-Based Methods: Exemplar-based techniques can be viewed as a refined subset of patch-based methods, designed to handle larger missing areas with complex structures. These approaches identify the most representative patches (exemplars) from the known regions to fill gaps. For instance, [24] highlighted how methods like those by Herling and Broll [18] and Guo et al. [17] incorporated structural and semantic constraints to reconstruct challenging textures and objects effectively.

The selection process focuses on high confidence areas first, iteratively expanding the known region. The inpainting follows this priority:

$$\text{Confidence}(P_u) = \frac{\sum_{i,j} C(i,j)}{|P_u|}, \tag{2.5}$$

where $C(i,j)$ is the confidence value of each pixel and $|P_u|$ is the patch size. The method prioritizes areas with the highest confidence for filling.

Although these traditional methods performed well for their time, especially with small gaps and uniform textures, they struggle when faced with large missing regions or complex patterns. These challenges have driven the transition toward deep learning-based solutions, which leverage neural networks to learn more sophisticated representations of image data.

2.1.7 Challenges and Future Directions

Despite the significant progress made by generative models in image inpainting, several challenges remain. Ensuring the semantic coherence of inpainted regions, handling high-resolution images, and reducing computational overhead are ongoing areas of research. Future work may focus on developing more efficient training and inference algorithms, improving the interpretability of generative models, and exploring new applications in various domains.

Generative models continue to be a promising area of research, with the potential to revolutionize image inpainting and other computer vision tasks. Their ability to generate realistic and contextually consistent data samples makes them invaluable tools for addressing complex inpainting challenges.

2.2 Mathematics behind the Denoising Diffusion Probabilistic Models (DDPM)

Denoising Diffusion Probabilistic Models (DDPM) are a class of generative models that learn to model the conditional distributions of data under sequential levels of diffusion. In our notation, we define x_0 as the original, clean data sample (i.e., the lowest temperature sample), while x_T denotes the final state of the forward diffusion process (i.e., the highest temperature sample, representing nearly pure Gaussian noise). Given an initial noisy sample, DDPM iteratively applies denoising steps to reduce the noise level until reaching the desired distribution. Formally, DDPM models the conditional distribution $p(\mathbf{x}_t|\mathbf{x}_0)$, where $\mathbf{x}_t$ represents the data at time step t during the reverse process that reconstructs x_0 from x_T.

During training, DDPM methods define a diffusion process that transforms an image x_0 into a noise distribution over T time steps. The final state x_T is modeled as Gaussian noise, $x_T \sim \mathcal{N}(0, \mathbf{I})$. The forward process adds Gaussian noise at each step t, transitioning from x_{t-1} to x_t using:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}). \tag{2.6}$$

where β_t is a variance schedule that controls the amount of noise added at each timestep t. β_t is typically chosen as a small, increasing sequence over T. for example:

$$\beta_t = \beta_{\text{min}} + t \cdot \frac{\beta_{\text{max}} - \beta_{\text{min}}}{T}, \tag{2.7}$$

with β_{min} and β_{max} as the minimum and maximum noise levels.

To simplify computations, we define α_t and its cumulative product $\bar{\alpha}_t$ as:

$$\alpha_t = 1 - \beta_t, \quad \bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s. \tag{2.8}$$

Here, α_t indicates how much of the original signal remains at each step, while $\bar{\alpha}_t$ represents the total remaining signal across all steps. These terms help simplify the forward process for efficient sampling.

The DDPM is trained to reverse this process. The reverse process is modeled by a neural network that predicts the parameters $\mu_\theta(x_t, t)$ and $\Sigma_\theta(x_t, t)$ of a Gaussian distribution. The reverse process of denoising is modeled by:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \tag{2.9}$$

The introduction of diffusion models, especially highlighted in the "RePaint" paper, represents a major breakthrough in this area. These models employ a pre-trained diffusion model and incorporate a defined mask during inference time to iteratively denoise images, enabling them to fill in gaps with remarkable accuracy and detail that align with the surrounding context.

2.3 UNet

UNet is a convolutional neural network (CNN) architecture initially proposed for biomedical image segmentation tasks. Its encoder-decoder structure, combined with skip connections, allows it to capture high-level contextual information. UNet has an image-to-image end and has been widely used in various fields, including

computer vision and spectrogram processing. It is capable of handling structured data and reconstructing missing regions with high accuracy.

UNet Architecture

The UNet [34] architecture is made of a symmetrical encoder-decoder design, and is named"UNet" due to its U-shaped structure. The encoder, or downward path, progressively reduces the spatial dimensions of the input through a sequence of convolutional layers followed by pooling operations. These convolutional layers start to detect simple features (e.g. edges) and gradually capture more complex features (like shapes and regions of the input image). This method brings a hierarchical representation of data to the model, which is essential in inpainting tasks. The decoder, or expanding path, mirrors the encoder but in reverse. It restores the spatial connections by upsampling the feature maps and combining them with corresponding feature maps from the encoder via skip connections. Skip connections will directly transfer the spatial details from the encoder to the decoder; This ensures if any detailed features are lost in the downward path, it will be recovered via skip connections.

Figure 2.4 illustrates the structure of UNet, highlighting its symmetrical structure and the use of skip connections to connect the encoder and decoder paths.

2.3.1 Application of UNet in Our Work

In this work, we adapt the UNet architecture for the task of spectrogram reconstruction, to address the challenge of inpainting corrupted regions. The input to the network is a 2-dimensional (time x frequency)spectrogram with masked regions representing missing or corrupted data, and the output is a reconstructed spectrogram with the gaps filled.

To incorporate UNet in our inpainting algorithm, we make the following modifications:

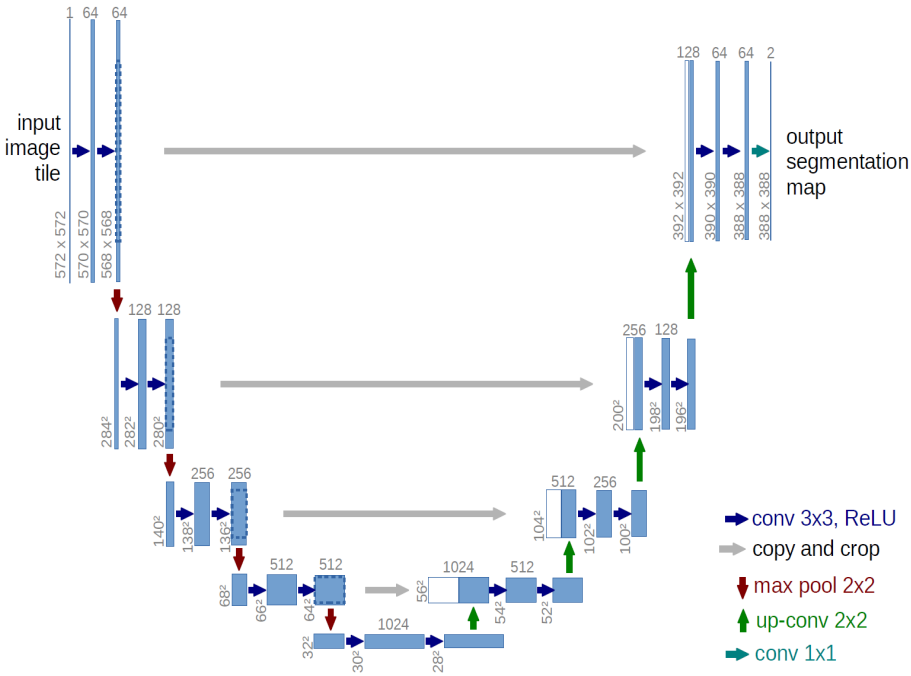

Figure 2.4: RePaint Diagram [29]

- The input layer is designed to handle spectrogram data with frequency and time dimensions, ensuring compatibility with the unique structure of the input.
- In some methods, the positional encoding is added to the input spectrogram, to provide the network with spatial context along the frequency axis, which will enhance its ability to learn frequency-dependent relationships.
- The number of layers and feature maps is adjusted to balance reconstruction quality and computational efficiency.

2.4 RePaint

The RePaint approach [29] for inpainting involves conditioning on known regions of the image. The inpainting process begins with initializing x_T from a normal distribution $\mathcal{N}(0, I)$. The algorithm then iterates from the last timestep T down

to 1. During each timestep, the image x_t is processed to estimate x_{t-1}. Authors assume that in each step we have a known and an unknown part of the final image, denoted by x_t^{known} and $x_t^{unknown}$ respectively.

The known part x_{t-1}^{known} is sampled using:

$$x_{t-1}^{known} \sim \mathcal{N}(\sqrt{\alpha_t}x_0, (1-\alpha_t)I).$$

Here, α_t is a factor that controls how much the starting or known state of the image, and x_0, affects the process as the algorithm works backward in time.

The unknown part $x_{t-1}^{unknown}$ is computed based on:

$$x_{t-1}^{unknown} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}\epsilon_\theta(x_t, t)\right) + \sigma_t z,$$

where $\epsilon_\theta(x_t, t)$ is the predicted noise for timestep t provided by the neural network, and z is sampled from $\mathcal{N}(0, I)$.

The final step for calculating x_{t-1} combines known and unknown regions:

$$x_{t-1} = m \odot x_{t-1}^{known} + (1-m) \odot x_{t-1}^{unknown},$$

In RePaint,m is a binary mask that indicates known (unmasked) and unknown (masked) regions. The algorithm involves two loops: an outer loop iterates over all timesteps T, while an inner refinement loop updates the image at each timestep. During the inner loop, the model refines the masked regions by leveraging the known context and the predicted noise. This iterative process ensures better estimation of the unknown regions as the model iteratively denoises the image. After completing the outer loop, the reconstructed initial image x_0 is returned as the final output.

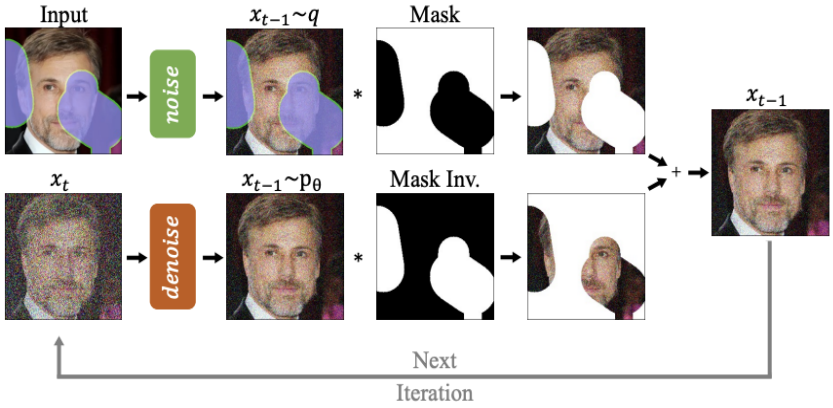

Figure 2.5: RePaint Diagram [29]

2.5 RFI and its Mitigation Techniques

As mentioned in the previous chapter, Radio Frequency Interference (RFI) poses a significant challenge in radio astronomy. RFI can originate from various sources, both internal (generated by instruments) and external (man-made radio emissions). The impact of RFI on astronomical observations is substantial, potentially corrupting weak cosmic signals and altering the integrity of collected data.

There are several works that explore specific techniques to restore corrupted data affected by RFI.

2.5.1 CLEAN Algorithm

The CLEAN algorithm [23] is a widely used deconvolution method designed to extract and reconstruct real astrophysical signals from radio interferometric data. In the context of HERA data, CLEAN is applied for Radio Frequency Interference (RFI) mitigation, based on the assumption that RFI is highly localized, in contrast to the spatially extended cosmic signal. The algorithm iteratively identifies and subtracts the brightest points, or "clean components," in the observed data. These components are treated as point sources and are iteratively convolved with the

telescope's point spread function (PSF) to construct a model of the true signal. The process involves the following steps:

1. Identify the brightest point in the data and assume it corresponds to a true source.
2. Subtract a scaled version of the PSF centered at this point from the data.
3. Record the position and intensity of this source (the "clean component").
4. Repeat the process until the residual data is reduced to noise levels.
5. Add back the clean components convolved with an idealized PSF to reconstruct the final image.

This iterative approach ensures that noise and interference are minimized while preserving the true signal. CLEAN is effective for mitigating RFI by treating it as a bright contaminant, though it assumes that RFI is sparse and does not overlap significantly with the cosmic signal. Its limitations include difficulty in handling diffuse RFI or overlapping sources, which has led to the development of modified versions.

2.5.2 Other RFI Mitigation Techniques

Beyond CLEAN, several other techniques have been explored for mitigating RFI, each with unique strengths and applications:

Least Square Spectral Analysis (LSSA): This method fits a model spectrum to the observed data using least squares minimization [27]. By selecting and removing spectral features associated with RFI, LSSA can effectively clean the data while preserving the actual astronomical signal.

Gaussian Process Regression (GPR): GPR [14, 25] models the data as a combination of a smooth signal and RFI using a probabilistic framework. By leveraging correlations in the data, GPR can separate the smooth background from high-frequency interference. However, it requires a careful choice of kernel functions to balance signal and noise.

Discrete Prolate Spheroidal Sequences (DPSS): DPSS [11,37] are used to isolate specific frequency bands affected by RFI. These sequences provide optimal spectral concentration and are suitable for band-limited RFI removal.

2.5.3 Applications in HERA and Other Observatories

In the HERA data analysis pipeline [33], authors have used a modified version of CLEAN tailored to their inpainting requirements. Moreover, they investigated other techniques including Least Square Spectral Analysis (LSSA), Gaussian Process Regression (GPR) [14, 25], and Discrete Prolate Spheroidal Sequences (DPSS) [11, 37]. These approaches leverage uncorrupted data to construct a basic model for the corrupted data, which is then substituted into the RFI-flagged regions. This can help mitigate the impact of RFI on the spectrum. However, the restored data may introduce potential errors in the analysis.

Chapter 3

Methodology

3.1 Overview of methods

We present our approach to address the inpainting problem with corrupted data. We have developed three primary methods, each building upon the previous one, to enhance our model's capability in reconstructing missing or corrupted image regions. Before delving into the details, we begin with a simple baseline. Our minimal baseline assumes the dataset is not corrupted. While this assumption is problematic, it provides a starting point. We will now discuss how we define the training and inference-time (inpainting-time) details for each method.

3.1.1 Method 1

Method 1: Training with Masked Data In this method, the model is trained using only the unmasked (observed parts) regions of the training data. The goal is to ensure that the model learns to have a valid prediction for the uncorrupted parts. The loss is penalized on the observed parts of the spectrogram. The RePaint algorithm was used during sampling, leveraging the model trained on clean data to refine the prediction in each step. This approach showed some limitations:

- The model can overfit to the clean regions without sufficient incentive to predict masked regions accurately.
- It treats the mask as independent parts of the image which will not incorporate the actual correlation values with its surroundings.

Algorithm 1 Method 1: Training with Masked Data

Require: Training data x_0, binary mask m

1: **Definition of** m: m is a binary matrix of the same shape as x_0, where:

$$m = \begin{cases} 1 & \text{for observed (unmasked) parts} \\ 0 & \text{for unobserved (masked) parts.} \end{cases}$$

2: **repeat**

3: Sample clean input $x_0 \sim q(x_0)$

4: Apply mask to isolate unmasked regions:

$$x_{\text{unmasked}} = m \cdot x_0$$

5: Predict $\hat{x}$ using the model:

$$\hat{x} = \text{Unet Model}(x_{\text{masked}})$$

6: Compute loss over unmasked regions:

$$\mathcal{L} = \|m \cdot (x_0 - \hat{x})\|^2$$

7: Update model weights θ according to the loss function

8: **until** Convergence

Algorithm 2 Inpainting algorithm for Method 1

Require: Mask m.

$x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ **do**
  for $u = 1, \ldots, U$ **do**
    $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ **if** $t > 1$, **else** $\epsilon = \mathbf{0}$
    $x_{t-1}^{\text{known}} = \sqrt{\bar{\alpha}_t} x_0 + (1 - \bar{\alpha}_t)\epsilon$
    $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ **if** $t > 1$, **else** $z = \mathbf{0}$
    $x_{t-1}^{\text{unknown}} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z$
    $x_{t-1} = m \odot x_{t-1}^{\text{known}} + (1 - m) \odot x_{t-1}^{\text{unknown}}$
    if $u < U$ **and** $t > 1$ **then**
      $x_t \sim \mathcal{N}(\sqrt{1 - \beta_{t-1}}\, x_{t-1}, \beta_{t-1}\mathbf{I})$
    end if
  end for
end for
return x_0

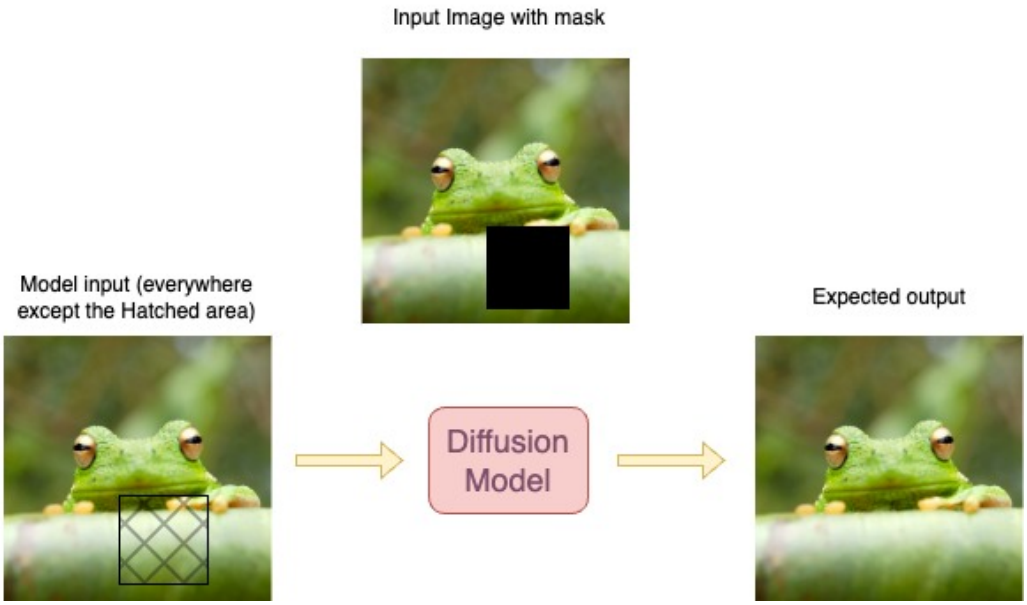

Figure 3.1: Method 1 simple form: We penalize the model output on parts of images where there is no hatching

Algorithm Explanation The inpainting process begins by initializing the image x_T with Gaussian noise. The algorithm then performs a reverse diffusion process, iterating from time step T down to t = 1. At each time step t, the algorithm executes U refinement iterations to enhance the inpainting quality. This loop adds noise back to the data during certain iterations rather than strictly removing it. This step helps the algorithm to achieve a coherent image as it prevents overfitting to a poor initial prediction for the masked regions.

1. **Noise Sampling:** For each iteration, noise ϵ is sampled from a standard normal distribution if $t > 1$; otherwise, it is set to zero to finalize the denoising.

2. **Known Regions Update:** The known regions of the image are updated using the original image x_0 and the sampled noise ϵ, scaled by the diffusion parameters.

3. **Unknown Regions Prediction:** The algorithm predicts the unknown regions by denoising x_t using the trained Unet $\epsilon_\theta(x_t, t)$, and adjusts it with the diffusion parameters, and adds additional noise z for stochasticity.

4. **Image Reconstruction:** The updated known and unknown regions are combined using the mask m to form the image x_{t-1} for the next iteration.

After completing all iterations and time steps, the algorithm outputs the final inpainted image x_0.

3.1.2 Method 2

Method 2: Improved Inpainting with additional fake masks We identified a limitation with Method 1: the denoising network had no incentive to focus on the masked regions of the input and fill them. To address this, we introduced fake random masks. The model is penalized for errors in denoising these fake masked regions, where ground truth is available. As the model cannot distinguish

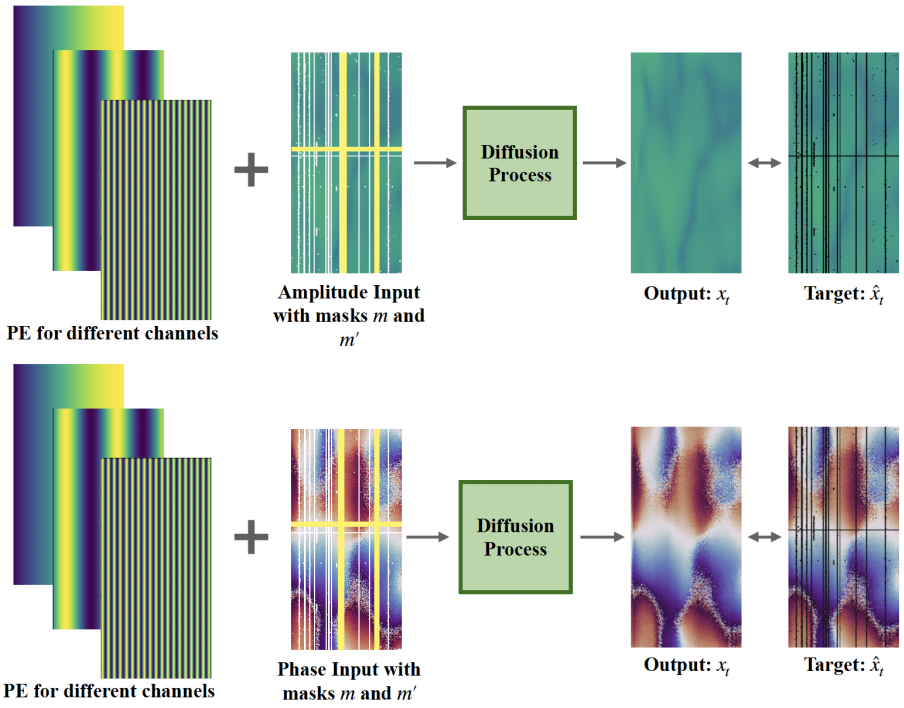

Figure 3.2: Illustration of the amplitude and phase of a spectrogram sample. We input the positional encoding along with the mask amplitude and phase into the diffusion process. Two sets of masks are used: m (pre-existing mask in black) and m' (additional masks in yellow). The U-Net diffusion model is trained by computing $\nabla_\theta(1-m')\alpha l_1 + (1-\alpha)|x_t - \hat{x}|^2$ where $l_1 = \|(1-m)x_t - (1-m)\hat{x}_t\|_1$..

between real and fake masks, it attempts to denoise all masked regions, improving overall performance. This fake mask is shown by m' in the algorithms and it has the same distribution as the real mask. Moreover, we ensured that these masks do not overlap with each other.

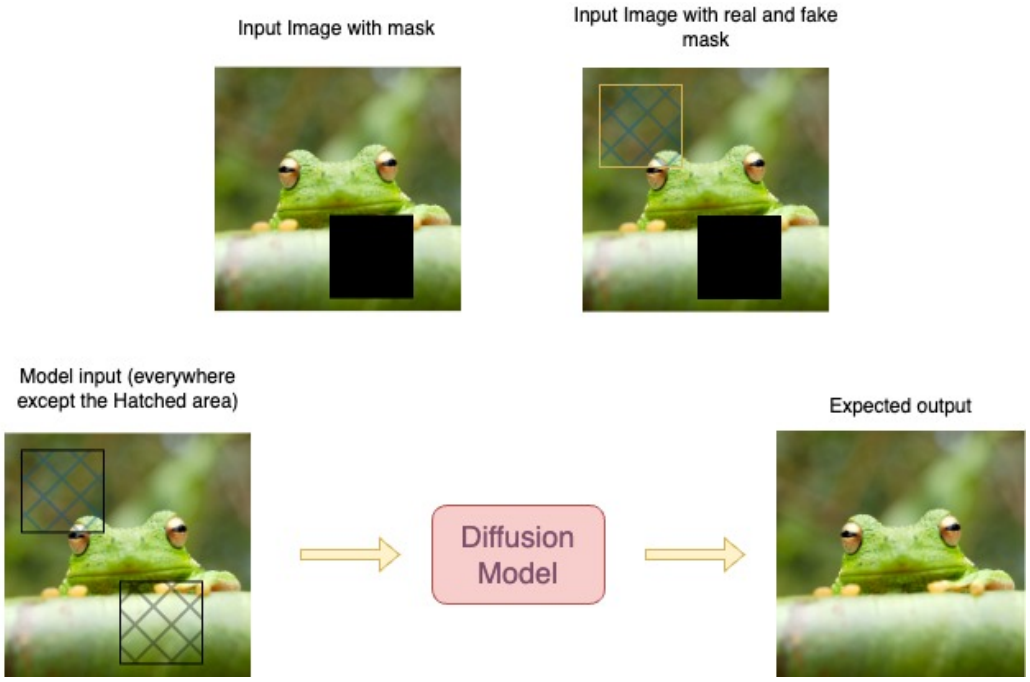

Figure 3.3: Method 2 simple form: We add fake makes on top of the real one, and not telling the model which is which. Then, we define our loss based on the hatches we know the ground truth of, and the intact part where there is no mask.

Algorithm 3 Training with Mixed Masking

Require: Training data x_0, real mask m, fake mask m', weight factor α

1: **repeat**

2: Sample clean input $x_0 \sim q(x_0)$

3: Generate real mask m, where:

$$m = \begin{cases} 1 & \text{for observed parts} \\ 0 & \text{for unobserved parts.} \end{cases}$$

4: Generate fake mask m' randomly, where:

$$m' = \begin{cases} 1 & \text{for randomly selected unobserved parts} \\ 0 & \text{otherwise.} \end{cases}$$

5: Apply masks to simulate input:

$$x_{\text{masked}} = m \cdot x_0 + m' \cdot x_0$$

6: Predict $\hat{x}$ using the model:

$$\hat{x} = \text{Model}(x_{\text{masked}})$$

7: Compute loss for both real and fake masks:

$$\mathcal{L} = \alpha \|m \cdot (x_0 - \hat{x})\|^2 + (1 - \alpha)\|m' \cdot (x_0 - \hat{x})\|^2$$

8: Update model weights θ using loss function

9: **until** Convergence

Algorithm Explanation The training algorithm follows the same structure as the training algorithm for method 1. The only difference is introducing and applying the fake masks. As mentioned before, this fake mask is randomly generated and applied to the training input. Fake masks introduce a similar distribution to the real one. While training, the model does not distinguish between real and fake masks; however, the critical point is that for fake masks, the ground truth is available, allowing the model to learn to fill them with exact ground truth values.

Algorithm 4 Inpainting algorithm for Method 1

Require: Mask m.

$x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ **do**
  for $u = 1, \ldots, U$ **do**
    $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ **if** $t > 1$, **else** $\epsilon = \mathbf{0}$
    $x_{t-1}^{\text{known}} = \sqrt{\bar{\alpha}_t} x_0 + (1 - \bar{\alpha}_t)\epsilon$
    $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ **if** $t > 1$, **else** $z = \mathbf{0}$
    $x_{t-1}^{\text{unknown}} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z$
    $x_{t-1} = m \odot x_{t-1}^{\text{known}} + (1 - m) \odot x_{t-1}^{\text{unknown}}$
    if $u < U$ **and** $t > 1$ **then**
      $x_t \sim \mathcal{N}(\sqrt{1 - \beta_{t-1}}\, x_{t-1}, \beta_{t-1}\mathbf{I})$
    end if
  end for
end for
return x_0

The inpainting algorithm for Method 2 follows the same structure as the inpainting algorithm described in Method 1.

3.1.3 Method 3

Method 3: Consistency Between Training and Sampling A key limitation we identified in Method 2 is the misalignment between the training and sampling processes. During training, the model learns to reconstruct masked regions (including fake ones and real ones) while noise is consistently added to these regions. However during the generation with RePaint, the network will be receiving unmasked inputs; this is different from its training, where parts of the input were masked, and only had the gaussion noise corresponding to that timestep. In this method, we apply the artificial mask m' to the model input along with the correct amount of noise to ensure that the input during sampling resembles the data seen by the model during training. The key addition is the inclusion of x'_t, which is computed in **line 7** of the RePaint algorithm:

$$x'_t = x_t \odot m' + (1 - m')(1 - \bar{\alpha}_t)\epsilon,$$

where x'_t has the artificial mask m' to appropriately introduce noise in the masked regions. This ensures that regions where $m' = 1$ remain untouched, while regions where $m' = 0$ are adjusted to include the correct level of noise $(1-\bar{\alpha}_t)\epsilon$. In addition, m' is designed such that regions where $m = 0$ (real masked regions) and $m' = 0$ (fake masked regions) ideally do not overlap.

The x'_t is the updated input to mimic the training input in sampling time to the denoising function at each timestep t. This fake mask m' is produced solely to mimic the artificial masking used during training, and x'_t has no other role beyond being the input to the denoising function.

This alignment ensures that the model processes masked regions during sampling like training, which can help improve the reconstruction quality and coherence in the final output.

Algorithm 5 Inpainting algorithm for Method 3

Require: Mask m.

$x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ **do**
  for $u = 1, \ldots, U$ **do**
    $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ **if** $t > 1$, **else** $\epsilon = \mathbf{0}$
    $x_{t-1}^{\text{known}} = \sqrt{\bar{\alpha}_t} x_0 + (1 - \bar{\alpha}_t)\epsilon$
    $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ **if** $t > 1$, **else** $z = \mathbf{0}$
    $x_t' = x_t \odot m' + (1 - m')(1 - \bar{\alpha}_t)\epsilon$
    $x_{t-1}^{\text{unknown}} = \frac{1}{\sqrt{\alpha_t}} \left(x_t' - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t', t) \right) + \sigma_t z$
    $x_{t-1} = m \odot x_{t-1}^{\text{known}} + (1 - m) \odot x_{t-1}^{\text{unknown}}$
    if $u < U$ **and** $t > 1$ **then**
      $x_t \sim \mathcal{N}(\sqrt{1 - \beta_{t-1}}\, x_{t-1}, \beta_{t-1}\mathbf{I})$
    end if
  end for
end for
return x_0

Algorithm Explanation Our work applies positional encoding to spectrograms to address the lack of translation symmetry along the frequency axis. This is particularly important because:

- Different frequency bands in a spectrogram often have distinct characteristics.
- The relationship between adjacent frequency bands can vary across the spectrum.
- Some phenomena in radio astronomy are frequency-dependent, and their position in the spectrum is informative.

By incorporating positional encoding, we enable our model to distinguish between different frequency bands while processing them with the same convolutional filters. Also, the model learns the frequency-dependent patterns more effectively and captures the absolute frequency positions, which can be crucial for identifying specific astronomical phenomena.

3.1.4 Positional Encoding

In transformer architectures, such as those used in natural language processing [41], the model processes input sequences in parallel, which inherently lacks the sequential order information present in traditional recurrent models like RNNs [10] or LSTMs [21]. To address this limitation, **positional encoding** is introduced to inject information about the position of each token within the sequence. This mechanism lets the model capture the order of elements which is essential in structured data like spectrograms, where positional relationships are important.
Traditional sequence models process inputs sequentially, which inherently includes a sense of order. Without positional encoding, the models cannot distinguish between different arrangements of the same set of tokens, which is critical for tasks such as language translation, or spectrogram analysis, where the position of frequency bands conveys meaningful patterns.

Translational symmetry means that a feature or pattern in the data looks the same even if it is shifted to a different position. Convolutional Neural Networks (ConvNets) assume this symmetry because they use shared filters across the input, making them effective for tasks like image processing, where patterns (like edges or shapes) are consistent across the image.

However, spectrograms do not have this symmetry along the frequency axis. Each frequency band contains unique information, and the relationship between neighboring bands can change. This makes it difficult for ConvNets to capture the structure of spectrograms properly.

To solve this, we use positional encoding, which adds positional information to the model input for the frequency axis.

This allows the network to recognize differences between frequency bands and capture the unique patterns in spectrogram data. This can help to fix the challenges posed by the assumption of translational symmetry.

Positional Encoding Mechanism

Positional encoding adds positional information to the input embeddings. The commonly used method, as proposed by Vaswani et al. [41], employs a combination of sine and cosine functions of varying frequencies These functions encode positions into continuous spaces, which allow the model to learn relative positions. The periodic nature of these functions can support generalization for all the sequence lengths.

The positional encoding vector for a token at position pos is defined as:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \tag{3.1}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \tag{3.2}$$

where:

- pos is the position of the token in the sequence.
- i is the dimension index.
- d_{model} is the dimensionality of the embeddings.

These sinusoidal functions create unique encoding patterns for each position pos and each dimension i. By design, these encodings enable the model to infer relative positions. For any fixed offset k, the positional encoding PE_{pos+k} can be represented as a linear function of PE_{pos}. This feature makes it easier for the model to capture relationships between tokens at varying distances.

Also the constant ”10000” is a scaling factor that adjusts the frequency of the sinusoidal patterns, ensuring the positional encodings span a wide range of values across different dimensions.

Visualization of Positional Encoding

Figure 3.4 illustrates the positional encoding vectors for different positions and dimensions. The sinusoidal patterns enable the model to capture positional relationships effectively.

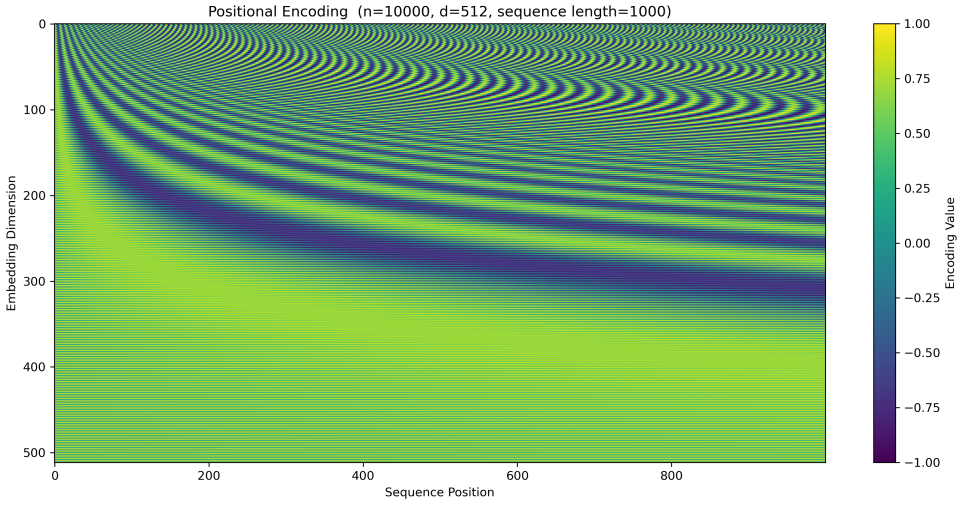

Figure 3.4: Illustration of Positional Encoding for Different Dimensions

Implementation Details

In practice, the positional encoding vectors are added to the input embeddings before they are fed into the model layers. This addition can be represented as:

$$\text{Input}_{\text{with PE}} = \text{Embedding} + PE \tag{3.3}$$

Addition operation is commonly used over concatenation because it is computationally simpler and does not introduce additional parameters, making it more efficient, especially in large-scale models.

Sinusoidal Positional Encoding for frequency

To incorporate spatial positional information into our model, we applied positional encoding to the spectrograms, a technique borrowed from Vaswani et al. [41]. Given a phase or amplitude tensor $x \in \mathbb{R}^{F \times T \times C}$ where $f \in 0, \ldots, F-1, t \in 0, \ldots, T$ and $c \in 0, \ldots, C-1$ represent frequency, time, and channels respectively, we compute positional encodings (independent of time) as follows:

$$\mathrm{PE}(f, t, c) = \sin\left(\frac{f \times \pi}{2F^{(c+1)/C}}\right) \tag{3.4}$$

This technique enables different channels to encode different frequencies, providing our model with an inherent understanding of spatial relationships within the spectrogram.

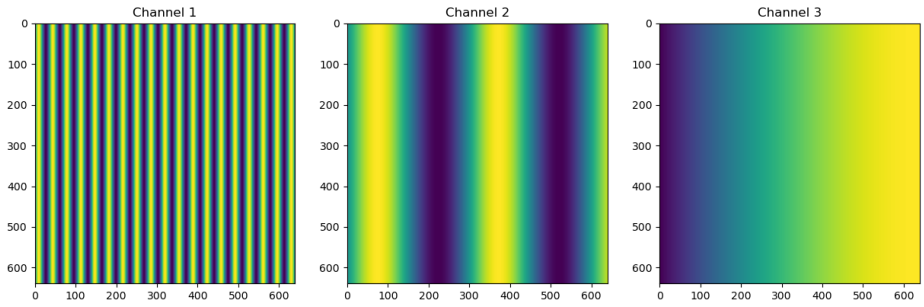

Figure 3.5: Visualization of sinusoidal positional encodings for a spectrogram. The x-axis represents frequency bands, the y-axis represents different encoding dimensions, and the color intensity represents the encoding values.

Specifically, we observed:

- Improved accuracy in reconstructing frequency-dependent features.
- Enhanced ability to distinguish between the RFI and genuine astronomical signals.

These improvements showcase the importance of providing the model with explicit positional information when dealing with structured data like spectrograms.

Chapter 4

Experimental Results and Discussion

4.1 Introduction to Experiments

In this chapter, we want to evaluate the effectiveness of our proposed algorithms on various datasets.

4.2 Datasets

4.2.1 CIFAR-10

The CIFAR-10 dataset consists of 60,000 color images of size 32×32 distributed evenly across 10 classes. We used this dataset as a benchmark to assess the effectiveness of our inpainting methods in scenarios with simple, colorful, and natural image data. Corruptions were introduced by applying random masks to the images, simulating missing or masked regions. After, we compared the results with the uncorrupted clean CIFAR images (baseline).

4.2.2 DermaMNIST

DermaMNIST is a dataset designed for skin lesion classification and related medical image analysis tasks [43]. It consists of 10,015 labeled dermatoscopic images categorized into seven classes representing various skin conditions, such as melanocytic nevi, basal cell carcinoma, and benign keratosis. Each image has been resized to 28×28 pixels. We picked this dataset to further validate the effectiveness of our algorithms.

4.2.3 Synthetic Spectrograms

The synthetic spectrogram dataset was generated using the Vispb software, simulating radio telescope outputs with controlled corruption. Each sample is a complex-valued tensor of size 640×480, representing the amplitude and phase of the frequency-time plane.

Below is a sample of the synthetic data showing the mask, which is divided into two parts. The white regions represent areas that are completely blocked, where meaningful predictions are not possible due to the large portions of frequency bands missing; therefore, these regions are not our focus. The red regions, however, indicate the differences in masks between nights, which vary from night to night. Unlike the permanent white mask that remains the same across all data points, the red regions are where we need to make an accurate prediction.

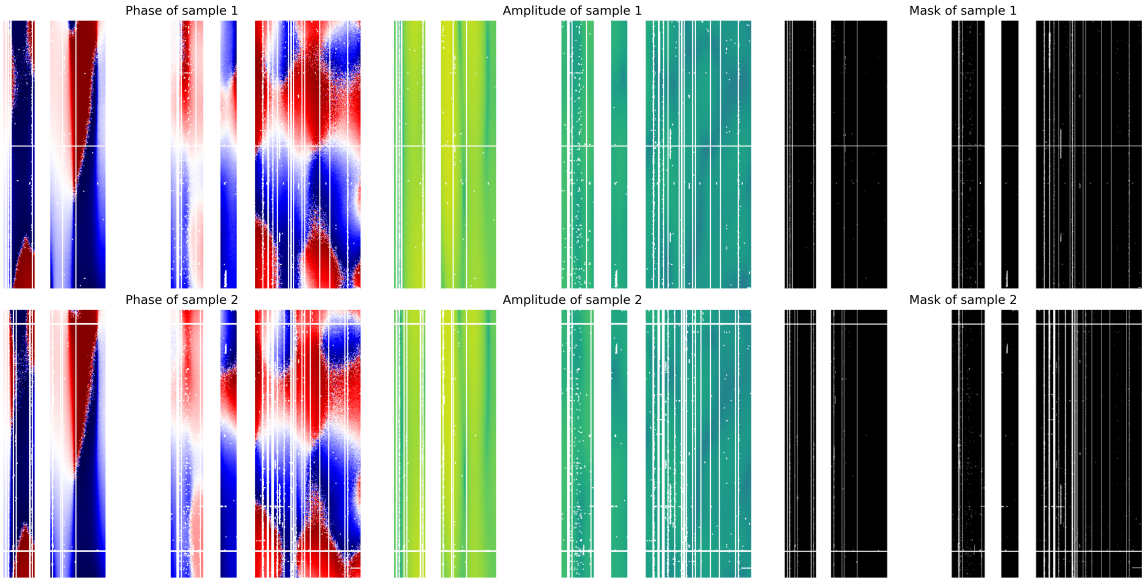

Figure 4.1: Visual comparison of phase, amplitude, and mask data across two data points.

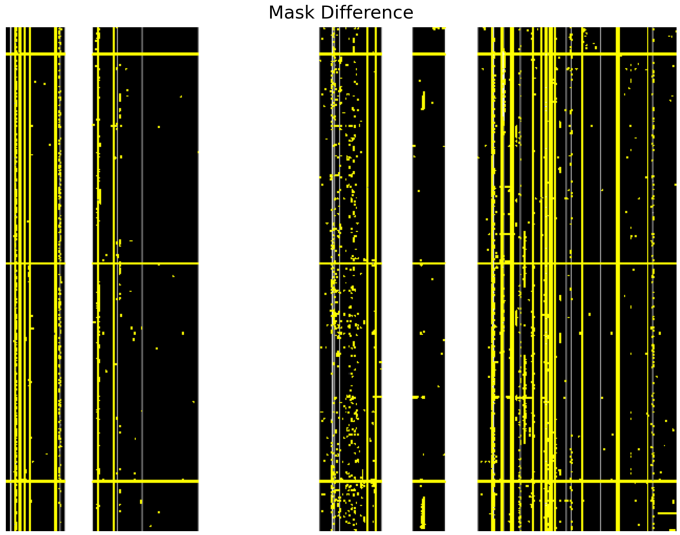

Figure 4.2: Mask Difference between two datapoints: Red parts indicate the narrow band masks we care to inpaint accurately

4.2.4 Real Spectrograms (HERA)

The HERA dataset includes real spectrograms from radio astronomy observations. Unlike synthetic spectrograms, the HERA data is inherently corrupted by Radio Frequency Interference (RFI), and no ground truth is available for the missing regions. To evaluate the quality of generated samples by the inpainting methods, the consistency of the reconstructed regions with the expected astrophysical signal was measured.

4.3 Experimental Setup

For all of the datasets, We utilized a U-Net architecture in the DDPM setup while training the model. To optimize the model's performance, we conducted a hyperparameter sweep, testing various combinations of learning rates, batch sizes, total loss weight coefficients, and the number of filters in each convolutional layer. The hyperparameter search was conducted as follows:

- **Learning Rates**: {0.0001, 0.00001, 0.000001}
- **Loss Weight contribution Coefficients (observed parts versus the masks)**: {0.3, 0.6, 0.9}
- **Batch Sizes**: {16, 32, 64, 128}
- **Number of Filters per Layer**: {64, 128, 256, 512}

Each combination was evaluated based on the validation loss, measured through the same number of epochs during training. The optimal set of hyperparameters was determined to reach the lowest validation loss before overfitting occurred. The final chosen hyperparameters were:

- **Learning Rate**: 0.001

- **Batch Size**: 32
- **Number of Filters per Layer in UNet**: 64, 128, 256, 512
- **Optimizer**: Adam.

These hyperparameter values were selected for the synthetic astrophysical spectrogram dataset. However, batch sizes were adjusted for other datasets to get the best results for each. For the MNIST and DermaMNIST datasets, we used a batch size of 256 to ensure the model processed more data in each batch, which resulted in faster convergence. For the real spectrogram dataset, a smaller batch size of 4 was chosen due to memory constraints when handling the larger and more complex data samples.

The data used in this study was in the complex number format, where both phase and amplitude information were preserved. For the synthetic data, we generated samples to simulate the corrupted spectrograms. The real dataset contained 259 samples and the synthetic dataset includes.

To adapt the UNet architecture for this data format, we adjusted the network layers and inputs accordingly. For Method 3, the positional encoding function of the frequencies was concatenated along with the 2D input, which included both the phase and amplitude components of the data. This addition helped the model leverage frequency-specific spatial context, improving the overall inpainting performance.

4.3.1 Evaluation Metrics

To evaluate the performance of our inpainting methods, we used two widely adopted metrics: Fréchet Inception Distance (FID) and Peak Signal-to-Noise Ratio (PSNR). We elaborate more by giving a summary of each score:

FID Score:

The FID score [19] measures the similarity between the distribution of reconstructed images and the ground truth images in a feature space learned by a pre-trained neural network. Lower FID scores indicate a closer match, with smaller values corresponding to higher visual fidelity in the reconstructed images.

PSNR:

The PSNR (Peak Signal to Noise Ratio), is a pixel-wise metric that quantifies the reconstruction accuracy by comparing the similarity between the reconstructed and ground truth images at a pixel level. It is expressed in decibels (dB), with higher PSNR values indicating better reconstruction quality. PSNR is especially useful for assessing datasets where ground truth data is available and pixel-accurate restoration is important.

For this study, we used the FID score for both of CIFAR10 and DermaMNIST datasets; These datasets were chosen because they contain ground truth images, and they include a high number of samples that can be sufficient for covering their distribution. their reconstruction is primarily focused on local pixel-level details rather than frequency-related structures. By using these metrics, we evaluated the inpainting quality in terms of both perceptual realism (via FID) and pixel accuracy (via PSNR).

4.4 Results

4.4.1 CIFAR-10

The CIFAR-10 dataset was used to evaluate the inpainting performance of the proposed methods on natural image data. Figure 4.3 illustrates sample inpainting results, showing the clean samples, the applied masks, and the outputs generated by various methods, including baseline, Simple RePaint, and the proposed approaches (Methods 1, 2, and 3). It is important to note that RePaint has been trained on much larger and clean datasets, such as Celeb-HQ and ImageNet. This

makes it an infeasible choice for the astrophysical applications of interest, as RePaint needs a clean (uncorrupted) and large dataset for its optimal practicability; in fields such as astrophysics clean and large-scale training data is often unavailable. Here, RePaint is included only for comparison purposes, as it provides an upper bound on the performance that can be achieved when training on corrupted data. This comparison helps to provide context for the results of our methods relative to a theoretically ideal scenario.

Table 4.1 reports the FID scores for each method. The baseline with no masking achieves the best FID score, as expected since it has access to the original unmasked images. The corrupted baseline demonstrates the highest FID score, indicating poor reconstruction quality. Among the proposed methods, Method 3 achieves a marginal improvement over Methods 1 and 2, which show its ability to generate realistic reconstructions while closely matching the data distribution.

Figure 4.3: Inpainted example from the CIFAR-10 dataset. On the left, the clean sample and the mask applied to it are plotted. We then show the final inpainted image using different methods.

Table 4.1: Inpainting results for CIFAR10

Method	FID Score
Baseline (No mask)	3.812
Corrupted Baseline	18.546
Method 1	3.834
Method 2	3.817
Method 3	3.820

4.4.2 DermaMNIST

For the DermaMNIST dataset, the performance metrics are summarized in the Table below. The baseline method means the model training happens on clean and unmasked data.

Table 4.2: Performance metrics for DermaMNIST dataset. The table reports PSNR and MSE for the three methods.

Method	Average PSNR	Average MSE
Baseline (model trained with no mask)	47.9335	1.9273
Corrupted Baseline (Model trained on masked images)	32.2578	4.5814
Method 1	46.7960	3.0519
Method 2	47.8519	2.2406
Method 3	47.8894	2.2165

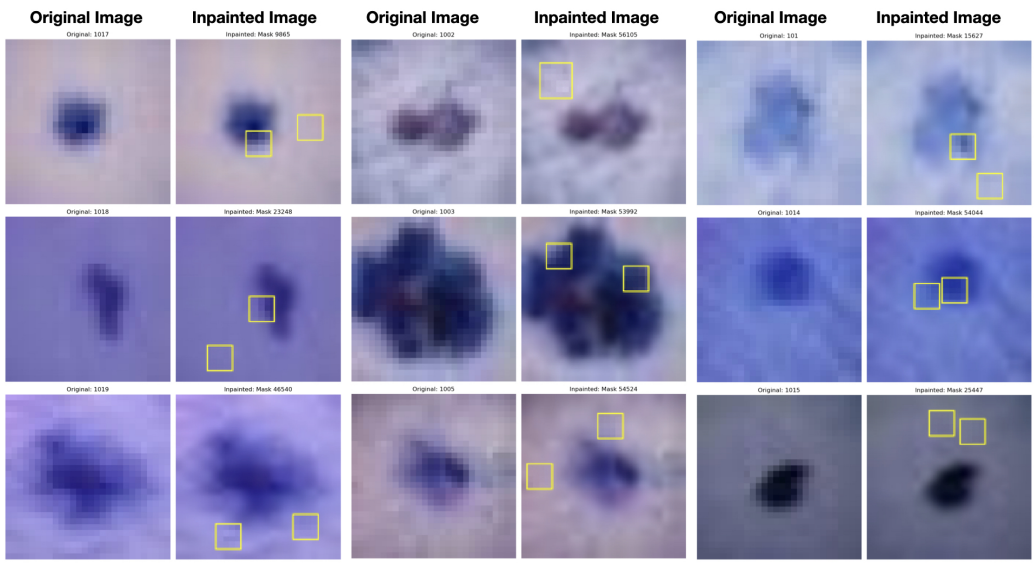

Figure 4.4: Visualization of results for the DermaMNIST dataset with method 3

4.4.3 Synthetic Spectrograms

For the Synthetic Spectrograms, figure 4.5 presents visualizations of the input, masks, and outputs from the baseline and proposed methods.

Table 4.3 highlights the reconstruction performance using PSNR. The baseline (model trained on data with no mask) method achieves the highest PSNR due to the absence of masking. Among the proposed methods, Method 3 with positional encoding (PE) outperforms the other ones, showing the role of positional encoding in improving frequency-sensitive reconstructions. These results show the effectiveness of incorporating frequency-specific patterns in spectrograms.

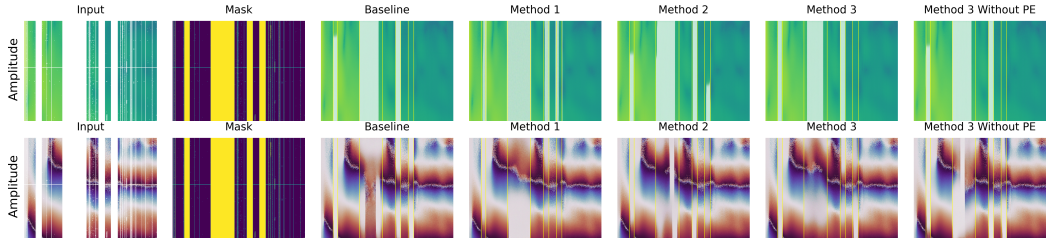

Figure 4.5: Visualization of results for the Synthetic dataset.

Table 4.3: Results for Synthetic spectrograms.

Method	Average PSNR
Baseline (No mask)	56.9176
Method 1	46.9958
Method 2	47.2915
Method 3	47.7325
Method 3 + PE	48.3748

4.4.4 Real Spectrograms (HERA)

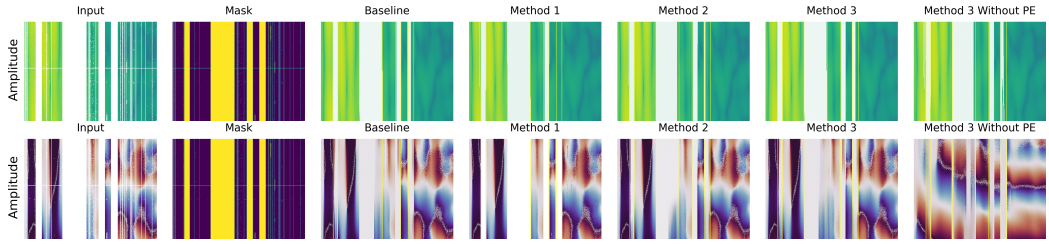

Figure 4.6: Visualization of results for the real astronomical dataset.

The HERA dataset (real data) was used to evaluate the inpainting performance of the proposed methods on data with complex frequency-dependent structures. Figure 4.6 shows the input, masked regions, and the reconstructions produced by each method.

4.5 Discussion

It's useful to note that although inpainting does not aim to recover the original, corrupted measurements, it helps to ensure smoother downstream processing. Instead of treating the masked regions as irretrievably lost, inpainting provides coherent reconstructions that reduce artifacts introduced by hard masking and aliasing. Thus, while these models do not increase the raw information content, they help preserve the structural and statistical coherency of the data, which is the im-

portance of this application in this application, and this can support more stable and accurate analysis.

Chapter 5

Conclusion

In this thesis, we explored the application of diffusion probabilistic models for inpainting corrupted spectrograms, particularly addressing the challenges caused by radio frequency interference (RFI) in astrophysical data. The problem was the unavailability of clean data in the spectrograms driven from satellites. To address this, we leveraged a diffusion model combined with embedded frequency positions to generate a smooth signal. Our approach builds upon an existing method for diffusion-based image inpainting. These proposed methods demonstrate improvements in both visual quality and quantitative metrics across various datasets.

One of the contributions of this work lies in proposing positional encoding to address and fix the model's assumption on the translational symmetry along the frequency axis of spectrograms. The positional encoding enables the model to differentiate between frequencies and exploit the inherent structure of spectrogram data. This approach is beneficial in domains that use spectrogram data, where frequency-dependent features often contain important information.

Through some experiments on datasets such as CIFAR-10, DermaMNIST, and synthetic spectrograms, we demonstrated the effectiveness of our methods. On synthetic spectrograms, the combination of positional encoding and Method 3 resulted in the highest PSNR scores, showing its ability to handle frequency-dependent

masking. Similarly, for CIFAR-10 and DermaMNIST datasets, our methods were competitive with the clean baseline (training on input with no mask), and also significantly outperformed the corrupted baseline (input with masks on top). This highlights the robustness of the proposed methods across different datasets.

In future work, we plan to validate the quality of the reconstructions by analyzing the power spectrum of the inpainted signals.

Bibliography

[1] J. Akeret and et al. Radio frequency interference detection using deep learning. *Monthly Notices of the Royal Astronomical Society*, 467(4):4800–4810, 2017.

[2] Coloma Ballester, Marcelo Bertalmio, Vicent Caselles, Guillermo Sapiro, and Joan Verdera. A variational model for filling-in gray level and color images. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 18(3):322–336, 2001.

[3] Cecilia Barnbaum and Richard F Bradley. Radio-frequency interference mitigation at the vlba and evla observatories. *Publications of the Astronomical Society of the Pacific*, 110(748):799–802, 1998.

[4] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. In *ACM Transactions on Graphics (ToG)*, volume 28, page 24. ACM, 2009.

[5] Marcelo Bertalmio, Guillermo Sapiro, Vicent Caselles, and Coloma Ballester. Navier-stokes, fluid dynamics, and image and video inpainting. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–I. IEEE, 2001.

[6] Ciprian Corneanu, Raghudeep Gadde, and Aleix M Martinez. Latentpaint: Image inpainting in latent space with diffusion models. In *Proceedings of the*

IEEE/CVF Winter Conference on Applications of Computer Vision, pages 4334–4343, 2024.

[7] Shaiyan Darabi, Eli Shechtman, Connelly Barnes, Dan B Goldman, and Pradeep Sen. Image melding: Combining inconsistent images using patch-based synthesis. In *ACM Transactions on Graphics (ToG)*, volume 31, page 82. ACM, 2012.

[8] J. P. Du Toit, J. S. Kenyon, and O. M. Smirnov. Deep learning for radio frequency interference detection: A comprehensive comparison. *Astronomy and Computing*, 42:100674, 2024.

[9] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1033–1038. IEEE, 1999.

[10] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

[11] Aaron Ewall-Wice, Nicholas Kern, Joshua S Dillon, Adrian Liu, Aaron Parsons, Saurabh Singh, Adam Lanman, Paul La Plante, Nicolas Fagnoni, Eloy de Lera Acedo, David R DeBoer, Chuneeta Nunhokee, Philip Bull, Tzu-Ching Chang, T Joseph W Lazio, James Aguirre, and Sean Weinberg. ¡tt¿dayenu:¡/tt¿ a simple filter of smooth foregrounds for intensity mapping power spectra. *Monthly Notices of the Royal Astronomical Society*, 500(4):5195–5213, October 2020.

[12] William T Freeman, Eric C Pasztor, and Owen T Carmichael. Learning low-level vision. In *International Journal of Computer Vision*, pages 25–47, 2000.

[13] Peter Fridman and Willem A Baan. Rfi mitigation using wavelet transforms. *Astronomy & Astrophysics*, 378(1):327–344, 2001.

[14] Abhik Ghosh, Florent Mertens, Gianni Bernardi, Mário G. Santos, Nicholas S. Kern, Christopher L. Carilli, Trienko L. Grobler, et al. Foreground modelling via Gaussian process regression: an application to HERA data. *Monthly Notices of the Royal Astronomical Society*, 495(3):2813–2826, 2020.

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[16] Asya Grechka, Guillaume Couairon, and Matthieu Cord. Gradpaint: Gradient-guided inpainting with diffusion models. *Computer Vision and Image Understanding*, 240:103928, 2024.

[17] Yanhong Guo, Jian Liu, Zhen Liu, Jian Zhang, and Deng Cai. Patch-based image inpainting with generative adversarial networks. *arXiv preprint arXiv:1708.06743*, 2017.

[18] Jan Herling and Wolfgang Broll. Real-time image-based information hiding using appearance-preserving rendering. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 207–212. IEEE, 2014.

[19] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, pages 6626–6637, 2017.

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020.

[21] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[22] Hui Huang, Kai Xu, Ralph R Martin, and Shi-Min Hu. Image completion using planar structure guidance. *ACM Transactions on Graphics (ToG)*, 33(4):129, 2014.

[23] J. A. Högbom. Aperture Synthesis with a Non-Regular Distribution of Interferometer Baselines. *Astronomy and Astrophysics Supplement Series*, 15:417, 1974.

[24] Asif Jam, Shoaib Mir, and Roshan Mir. A survey on deep learning techniques for image inpainting. *Visual Informatics*, 5(2):50–61, 2021.

[25] Nicholas S Kern and Adrian Liu. Gaussian process foreground subtraction and power spectrum estimation for 21cm cosmology. *Monthly Notices of the Royal Astronomical Society*, 501(1):1463–1480, December 2020.

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

[27] A. Leshem, A. J. van der Veen, and A.-J. Boonstra. Multichannel interference mitigation techniques in radio astronomy. *The Astrophysical Journal Supplement Series*, 131(1):355–367, 2000.

[28] Y. Li, Y. Zhang, and Y. Wang. Radio frequency interference mitigation and statistics. *Research in Astronomy and Astrophysics*, 21(5):1–10, 2021.

[29] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022.

[30] Michael Mesarcik, Albert-Jan Boonstra, Elena Ranguelova, and Rob V van Nieuwpoort. Learning to detect radio frequency interference in radio astronomy without seeing it. *Monthly Notices of the Royal Astronomical Society*, 516(4):5367–5379, 2024.

[31] Ali Mira, Bill Hursky, and Jafari Najmeh. Radio frequency interference (rfi) in radio astronomy: A detailed study. *ResearchGate*, 2023.

[32] A. Mohammed, W. Baan, and A. Fridman. Rfi mitigation techniques in radio astronomy. *Journal of Astronomical Instrumentation*, 10(3):1–25, 2021.

[33] Michael Pagano et al. Characterization of inpaint residuals in interferometric measurements of the epoch of reionization. *Monthly Notices of the Royal Astronomical Society*, 520(4):5552–5572, 2023.

[34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[35] Stefan Roth and Michael J Black. Fields of experts: A framework for learning image priors. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2:860–867, 2005.

[36] Chitwan Saharia et al. Palette: Image-to-image diffusion models, 2022. arXiv preprint.

[37] David Slepian. Prolate spheroidal wave functions, Fourier analysis, and uncertainty—V: The discrete case. *Bell System Technical Journal*, 57(5):1371–1430, 1978.

[38] Jascha Sohl-Dickstein, Eric A Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[39] Interference Technology. Identifying and locating radio frequency interference (rfi). *Interference Technology*, 2022. Accessed: 2024-07-28.

[40] David Tschumperle and Rachid Deriche. Fast anisotropic smoothing of multi-valued images using curvature-preserving pde's. *International Journal of Computer Vision*, 68(1):65–82, 2005.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[42] Yifan Yan, Yifan Zhang, Yujia Chen, Zongyu Huang, Shuang Wang, and Shijie Xu. Latentpaint: Image inpainting in latent space with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–10. IEEE, 2023.

[43] Jiancheng Yang, Rui Shi, Donglai Wei, Zequn Liu, Lin Zhao, Bilian Ke, Junjie Xia, Yong Cai, Yao Zheng, Hongming Ren, et al. Medmnist: A lightweight benchmark for biomedical image classification. *arXiv preprint arXiv:2010.14925*, 2020.

[44] Shiyuan Yang, Xiaodong Chen, and Jing Liao. Uni-paint: A unified framework for multimodal image inpainting with pretrained diffusion model. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 3190–3199, 2023.

[45] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the*

IEEE conference on computer vision and pattern recognition, pages 5505–5514, 2018.