SAPIENZA
UNIVERSITÀ DI ROMA

Diffusion Models for Generating and Augmenting Earth Observation Data: exploring a few use-cases

Facoltà di Ingegneria informatica automatica e statistica

Corso di Laurea Magistrale in Artificial Intelligence and Robotics

Candidate

Fulvio Sanguigni
ID number 1797953

Thesis Advisor

Prof. Irene Amerini

Co-Advisor

Dr. Bertrand Le Saux

Academic Year 2022/2023

Thesis defended on 21 July 2023
in front of a Board of Examiners composed by:

Prof. Massimo Mecella (chairman)
Prof. Simone Agostinelli
Prof. Irene Amerini
Prof. Danilo Comminiello
Prof. Gabriele Proietti Mattia
Prof. Simone Lenti
Prof. Simone Scardapane

Diffusion Models for Generating and Augmenting Earth Observation Data: exploring a few use-cases
Master's thesis. Sapienza – University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: sanguigni.1797953@studenti.uniroma1.it

To my beloved family for supporting me in everything I do. To my brother for the countless days spent together this year. To the colleagues and friends from ESA, for the growth I had in these months.
To my Grandfather.
Instinct choices, Rationality follows

Contents

Chapter 1

Introduction

Earth Observation (EO) is the discipline deputed to monitor, analyze, describe and predict the different physics phenomena happening on the Earth, ranging a wide and diverse timespan, from a daily basis to multiple years and decades of data collection. It's very important to capture the pulse of the planet, by monitoring the environment, showing the climate change to the world, and performing other several tasks, such as biomass estimation, atmosphere pollution indexes, precipitations levels, together with human activity supervision, such as landcover usage and population density.
This wide and diverse set of tasks is possible thanks to the numerous ESA missions that in the latest years launched the satellites, critical to provide data and insights for our EO works, starting from the first one, Meteosat, cryosat, envisat, spanning through the Sentinel Products, and finally extending them for the observation of other celestial bodies, such as the Juice mission.

The Two Revolutions in EO The constellation generated by ESA missions, together with the the data analysis performed on the ground, contributed to an important understanding of the causes and effects explaining the life of our planet. But there have been two major revolutions in the last fifteen years:
Firstly, the remote sensing community was overwhelmed by a data avalanche, thanks to the availability of new sensors and new constellations, such as the Sentinel products, launched in 2014, which provided lots of images accumulating years of observations. These data are mainly public, open-sourced for the scientific community through the Copernicus and sentinel-hub platforms; at the same time, there exist some private collections, mainly referring to constellations of low-cost Cubesat.
Secondly, we refer to the Artifical Intelligence (AI) revolution which happened in the last ten years, which contributed to give more powerful tools for the already existing tasks, created new possible applications for the existing ESA products, and finally leveraged the new technologies introduced in the first point of this revolution. In particular, AI has a special branch deputed to the analysis of images, namely Computer Vision. Many computer vision approaches have found a parallel application for EO tasks. Convolutional Neural Networks (CNNs) contributed to the classification of objects, and they have been succesfully translated for landcover classification or cloud detection in an image; at the same time, Deep Neural Networks (DNNs) were mainly adopted for regression tasks, and EO scientist adapted them for the Biomass estimation problem. Finally, Recurrent Neural Networks (RNN) contributed to the prediction of future events leveraging time series analysis of previous ones; in EO

they have proven critical for weather forecasting and wildfire predictions.

Current Challenges and Criticities We can identify two main challenges from this situation: Firstly, all the important AI revolutions and new data availability came at a cost, because AI approaches need humans scientists validating and labeling the acquisitions from the space. This problem arises because many approaches are supervised, so they need to be trained on an image and the corresponding label to properly learn the task and applying it to new, unobserved data. This implies changing the usage paradigm of AI, from leveraging labels to generate them.
ESA products have a lot of labeled data, actually many more of the labeled ones. The recent advancements in Self-Supervised learning are an important step towards the direction of leveraging unalebeled data, but they do not solve the problem of scarce label availability; we will need models leveraging both.
Secondly, CNNs are still used in EO tasks which are more demanding than simple classification; in this case, we refer to cloud removal or image super-resolution (more generally, image enhancement), which require a model of the image data distribution. This is suboptimal with discriminative architectures, such as CNNs; for this reason, we will focus on the adoption of generative AI models. This conclusion introduces another challenge: can we find some generative models for the specific task we have in mind?

A New Way of Generating Data Previous generative AI works where based on Generative Adversarial Networks (GANs). Even though they are valid architectures to generate to EO data (and labels), they still have some hidden problems, such as stability in training and mode collapse. For this reason, we explore a new, promising direction in the generative domain, namely Diffusion Models, a recent architecture which solves the underlying problems of GANs.
Given the promising features of these models, we will show in the rest of the work how to leverage them properly for the Remote Sensing domain. Moreover, this need rises other two subquestions:
First, which kind of EO subcases can be adopted to show the potential of diffusion models?
Second, for these use-cases, is it possible to build a proof-of-concept using diffusion models?
The main contributions of this work are:

1. We provide an application to cloud removal, providing an assessment of the potential related to this model

2. We provide an application to urban re-planning scenarios, to show the potential impact of human presence and absence on the environment

3. We provide a new data preparation pipeline for downstream tasks, such as change detection. We will focus on the label generation problem here

We organize the thesis as follows:
In Chapter 2 we briefly present an overview of previous Generative Models, together with the Earth Observation background (available tools and previous EO-related works leveraging AI)
In Chapter 3 we present the complete diffusion model framework lying under the hood, starting from its mathematical foundations, covering the architecture extensions and variants, and finally giving an overview of related applications

In Chapter 4 we present an overview of our method, highlighting the main architecture, the mathematical concept, and our design choices
In Chapter 5 we show the different tasks we targeted for our diffusion models, together with some ablation studies concerning the choice of data and different architectures.
Finally, in Chapter 6 we summarize our findings, providing insights for the reader, final comments on the value of these models as well as on their pitfalls, and finally we leave suggestions and ideas for future improvements and extensions of our work.

Chapter 2

Related Works

2.1 Generative Models

Generative models have established as a new paradigm for computer vision tasks. They differentiate from classical discriminative architectures because they try to model the data distribution, instead of looking for categorization or boundaries. We will briefly present all the different approaches falling into this category of models.

2.1.1 Variational Autoencoder (VAE)

VAE [1] is one of the first attemp to generate new data out of a target distribution. They borrow the same structure of an autoencoder (AE) [2] with a different sampling procedure: AEs are trained to perfectly reproduce an image, for this reason they are common in anomaly detection. VAEs, instead, add gaussian noise to the latents extracted from the encoder part, and then the decoder is fed with this new representation. In formulas, we produce a new latent variable z from the mean and variance of original latent in this way:

$$z \sim q_{\mu,\sigma}(z) = \mathcal{N}(\mu, \sigma^2)\epsilon \sim \mathcal{N}(0,1)z = \mu + \epsilon \cdot \sigma \tag{2.1}$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ and $\mathbf{z} \sim \mathcal{N}(\vec{\mu}, \sigma^2\mathbf{I})$. This formula is known also as the *reparameterization trick* because it provides a final formula to sample a variable z from a given distribution q. VAEs are very important for diffusion models for two reasons: firstly, the variational inference formulation will be resumed for DDPM (and following works) formulation;
secondly, they will be used in combination with the U-Net for the image encoding part and guidance. Diving into the mathematical part, the most important trick of VAEs lie in the loss term formulation, which is an *Evidence Lower Bound*(ELBO):

$$\mathbb{L}_{\theta,\phi}(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}, \mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})] \tag{2.2}$$

This loss term allows to compute the difference between target distribution $\mathbf{x}$ and latent features $\mathbf{z}$.
VAEs have been widely extended and improved in following works, both to explore different applications and to mitigate some hidden problems, such as *mode collapse*. $\beta-$VAE [3] and VQ-VAE [4] have tried to propose a different concept and architecture for VAEs: the first one introduces a parameter β to weight the D_{KL} term in the loss; it is intended to regularize the generative capacity of the VAE, keeping the distance from the target distribution inside ϵ.

The second one realize a discrete version of the VAE latents, with a follow-up work, VQ-VAE2 [5] trying to combine VQ-VAEs and autoregressive models [6] by means of cascaded, multiresolution latent maps instead of just one.

2.1.2 Generative Adversarial Network (GAN)

GANs [7] have been for long time the benchmark to beat in image synthesis. The quality of generation led to several efforts to mitigate the DeepFakes videos [8],[9] arising on the web, as well as tracking back the origin of an image (*image forensics*) [10].
GANs are essentially based on a minmax loss between a *generator* part to synthetize a fake image, and a *discriminator* part to classify the image as real or fake, as highlighted in eq. 2.3

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{2.3}$$

In following works, GANs have been extended acting in two directions:

1. By modifying the loss term, in particular substituting the KL-divergence with other types of losses such as Wasserstein distance [11]

2. By tailoring the GANs framework to specific tasks, such as StyleGAN [12], or focusing on model size [13] or limited data amount [14]

2.1.3 Flow Based Models

Flow based Models [15] try to reverse back the starting probability distribution. They are usually compared with GANs avd VAEs for image generation, but at the moment they still lack sampling quality and enough flexibility. They are usually split between normalizing flows [15] and autoregressive flows [6] [16] [17].
Autoregressive flows estimate the conditional probability, therefore going backwards sampling instead of forward. As highlighted in [18], GANs and VAEs (plus optical flow) suffer of two, complementary problems: training stability and mode collapse. None of the two methods have been decisive in solving these two issues, therefore there is the need for further investigation on other methods

2.2 Earth Observation Background

In this section we are going to provide a comprehensive overview of the remote sensing domain; we will start by giving a brief introduction to the available tools providing both the raw data as well as the procedures necessary to preprocess them. Then, we will present a complete review of the current works levearing EO data using AI approaches, with focus on the main research questions we posed in the introduction.

2.2.1 Esa Products

The European Space Agency launched several missions in the past, with the goal of capture the activity of our planet with more detail and coverage. These efforts led to the current available satellites, everyone of them has some unique features to provide different data; the diversity provided by Esa products is fundamental to furnish valuable and rich insights to the researchers and the anaysts, both in

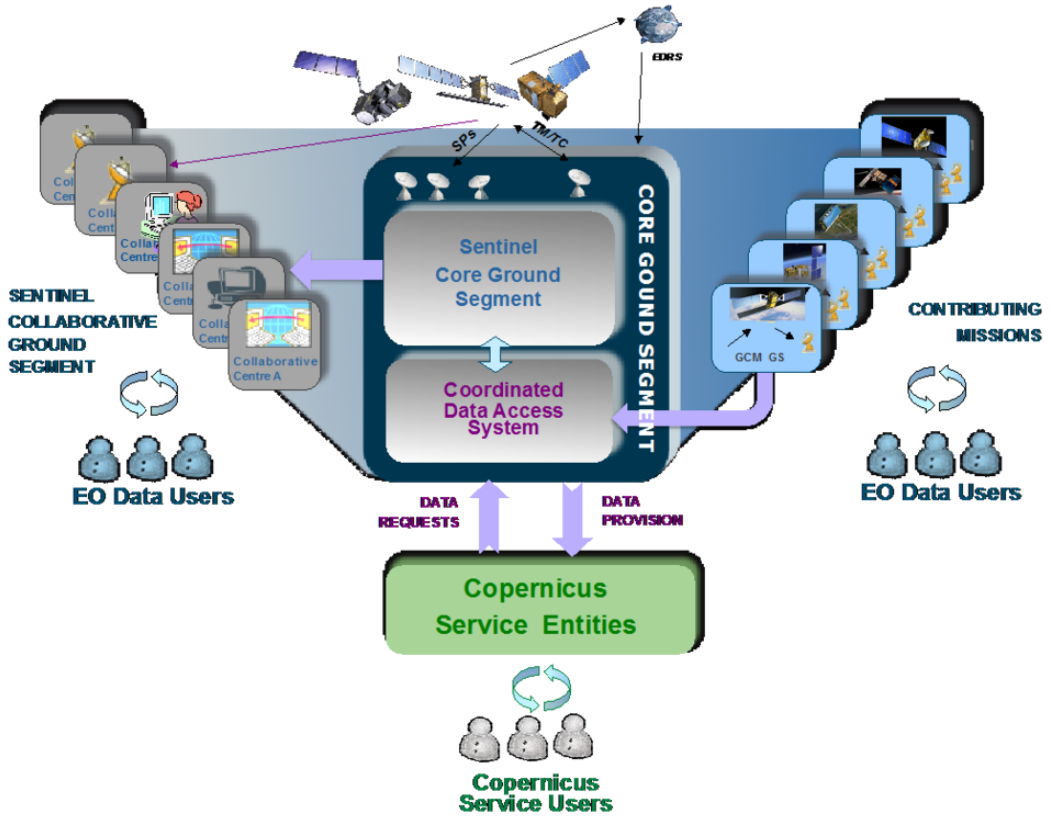


Figure 2.1. Esa data collection and analysis pipeline from Sentinel user guide

real-time and for post-processing operations. In the image below (2.1) we provide a general scheme for the data acquisition and usage at Esa In this context, we will start to deep dive into every single point in the list.
First, we will give an overview of the existing constellations acquiring images of the Earth Actually, there exist many different missions providing data for the scientists:

- **The Sentinel Products**: The first mission of Sentinel has been launched in 2014. Actually, Sentinel products comes from: Sentinel1, Sentinel2, Sentinel3, Sentinel 5p, Sentinel 4, Sentinel 5, Sentinel 6. For the interested reader, we leave two references, the Sentinel Website and the Esa website for a brief introduction to the functionalities of these products. The main products comes from Sentinel-1 and Sentinel-2. Sentinel-1 provides radar data, and they proved to be beneficial as an alternative to Sentinel-2 data in some challenging scenarios, such as presence of clouds.
  Sentinel 2, instead, provides 13 different bands to be extracted. Amongst these bands, we mention expecially the well known RGB bands. Every different frequency in Sentinel-2 products is beneficial for a specific task, because it offers data with very different spatial resolutions, spanning from the 10 m resolution of RGB (bands 2,3,4) to 60m (bands 1,9,10), depending if we want to focus on the details or if we want to elaborate a wide portion of territory

- **PRISMA**: PRISMA is a medium-resolution satellite developed and owned by Italian Space Agency (ASI). Esa and ASI collaborate to fully leverage the data. The full acquisition comes from an hyperspectral sensor and consists in a 66

channel image divided into two main bands, the VIS/NIR and the NIR/SWIR. The huge number of channels is due to the hyperspectral nature of this sensor, which provides much more wavelengths in the same spectra compared to a multisensory acquisition.

- **Φ-sat** is a very recent Esa mission deputed to bring AI onboard, testing real time the algorithms designed on the Earth
- **MeteoSat** is the first product launched by Esa; actually it's a constellation of 11 satellites, contributing to monitor the weather of our planet
- **Landsat series**: They are provided by Nasa, and Esa has an agreement to reuse and distribute the data for its activities. The latest satellite is Landsat-8, launched in 2013. It offers two types of sensors: the Operational Land Imager (OLI) and a Thermal Infrared Sensor (TIRS). The first one has a special focus on band 9 (new near-infrared, NIR) and band 1 (new deep blue channel), with the goal of detecting cirrus and coastal zones observations, offering spatial resolutions at 15m and 30m. The second one is a 100m thermal sensor with the goal of corregistering the OLI data on the ground

These products does not stay all in the same orbit. In fact, it's advantageus to use different type of trajectories in order to acquire data from a different perspective, or to have reproducible conditions. The current trajectories are:

- **Geostationary Orbit** (GEO). It's critical for every application needing a sensor pointing exactly at the same point in space in every moment. The interesting side effect of this configuration is the large distance from the earth (35 786km), thus enabling a very large Field of View (FOV)
- **Low Earth Orbit** (LEO) this is mostly used as a launchpad to carry satellites in the space or bringing instrument on board to the International Space Station. I's not very used due to the difficulty of tracking a fast runnign satelllite from the ground
- **Medium Earth Orbit** (MEO) is one of the most covered trajectories, from examples by the Galileo constellation.
- **Polar and Sun Syncronous Orbit**: in this case the object does not revolve around following the equator, but it passes the North and South Pole within a 0-30 degrees range. The Sentinel-2 products are covering this one. Its heigh range from 200 km to 1000 km.
- **Transfer Orbits and Geostationary Transfer Orbits**: these are particular types of orbits where the goal is to pass from one orbit type to another.
- **Lagrange points** these are Earth-Sun relative points, four times far from earth than the GEO trajectory. Actually, they proved very useful for missions needing to capture deep features in the space, without being disturbed by the natural radiance emitted by the Earth.

Now we will provide a description of the sensors acquiring data from the space, and how to threat this data. We will focus mainly on Sentinel 1 and Sentinel 2 acquisitions.
Sentinel 2 has a Multispectral Instrument onboard, returning 13 different bands as shown in table 2.2 The MSI sensor consist on two focal planes, a dichronic

	S2A		S2B		
Band Number	Central wavelength (nm)	Bandwidth (nm)	Central wavelength (nm)	Bandwidth (nm)	Spatial resolution (m)
1	442.7	20	442.3	20	60
2	492.7	65	492.3	65	10
3	559.8	35	558.9	35	10
4	664.6	30	664.9	31	10
5	704.1	14	703.8	15	20
6	740.5	14	739.1	13	20
7	782.8	19	779.7	19	20
8	832.8	105	832.9	104	10
8a	864.7	21	864.0	21	20
9	945.1	19	943.2	20	60
10	1373.5	29	1376.9	29	60
11	1613.7	90	1610.4	94	20
12	2202.4	174	2185.7	184	20

Figure 2.2. Values for the different bands of Sentinel-2 products, from Sentinel user guide

VNIR/SWIR beam splitter and two distinct arrays of 12 detectors mounted on each focal plane covering VNIR and SWIR channels respectively, as shown in fig. 2.3. A scheme of the extracted bands and their division can be seen in fig. 2.4. For further details we remind to the whole sentinel 2 technical guide For sentinel 1 we have

.

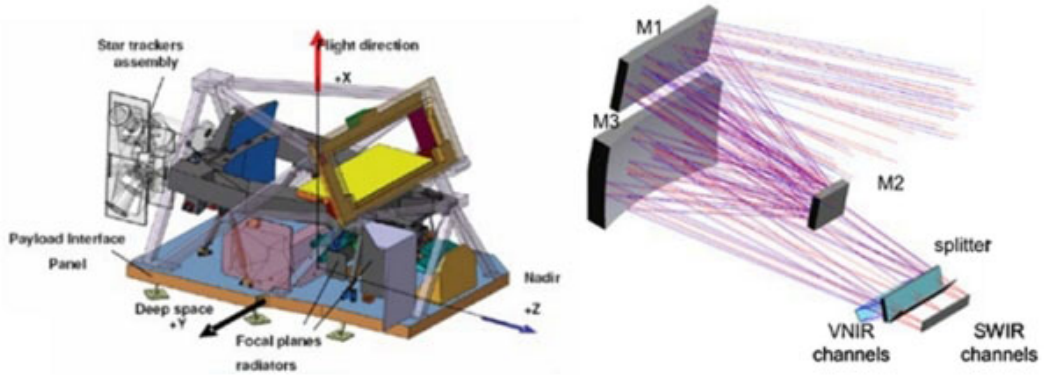


Figure 2.3. MSI sensor acquisition, image from Sentinel user guide

instead a Synthetic-Aperture-Radar sensor (SAR). The acquisitions are performed in all-weather and day/night. Moreover, the instrument can acquire data in four exclusive modes:

- Stripmap (SM): the antenna points to a fixed azimuth and elevation angle, while the ground illumination is performed with a series of pulses
- Interferometric Wide swath (IW): Data acquired in three swaths using a

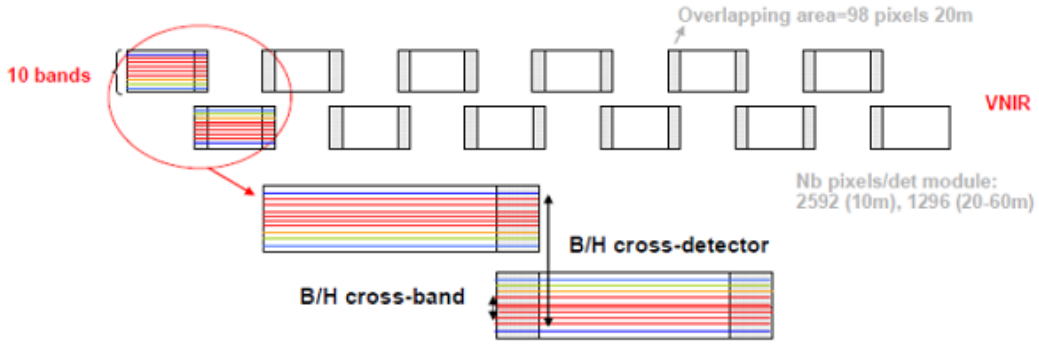


Figure 2.4. Overview of data spectral bands, see Sentinel user guide

Progressive Scanning SAR (TOPSAR) imaging technique. This mode requires to syncronize bursts time to time. Actually, it's the most common mode in SAR acquisitions.

- Extra Wide swath (EW): 5 different swaths acquired through the TOPSAR technique; in this case, it enphasizes a very large coverage at the expense of spatial resolution
- Wave (WV): In this case it has been used a particular technique, dividing a scene into smaller stritmap scenes called "vignettes"; they are usually acquired in pairs, one with a near incidence angle while the next one with a far range incidence angle. According to Copernicus technical guide, this is the most used strategy for open ocean mapping.

In fig. 2.5 we leave a summary of these 4 different acquisition modalities. We remember that every modality has different swath width, azimuth and elevation values, and polarization levels. For the technical details, we refers to the Sentinel-1 technical guide

Acquisition levels and Preprocessing of data For all the sentinel products, the general architecture is:

1. A ground segment deputed to both Sentinel mission control and raw data acquisition
2. Levels processing, as we will explain in the following lines
3. (variable) some postprocessing operations such as calibration of data, expecially referring to Sentinel-1 products

From previous paragraphs we can understand that the Sentinels are a diverse and rich source of data. As we show in fig. 2.6, only the latest ones are released to the end user. The rest of the levels are intermediate, preprocessing steps to carefully prepare a resusable and modular dataset for the scientific community. Level 0 operations are performed real-time to package the raw data into ordered and verified granuli; in particular, it performs some telemetry analysis for bit range check. Time information is added to a granulus through datation; afterwards, the image is extracted in low-resolution for quicklook generation, and finally the consolidation phase prepares level-0 data to be processed by level-1 operations.
As written on the technical guide for Sentinel 2, Level-2 is splitted into 3 categories:

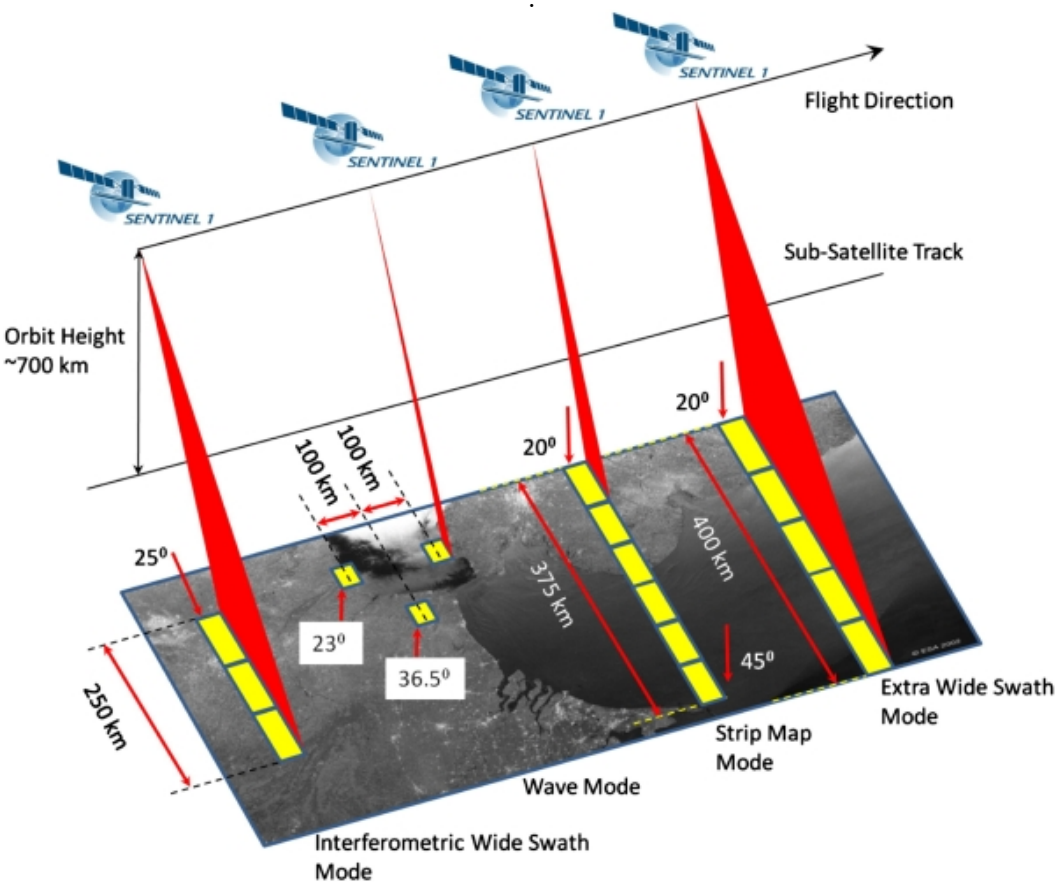


Figure 2.5. Sentinel 1 mode acquisitions, from Sentinel user guide

1. Level-1A processing is focused on decompressing relevant mission source packets. The Level-1A product is not available to Users.

2. Level-1B processing uses the Level-1A product and applies the required radiometric corrections. The Level-1B product is not available to Users.

3. Level-1C processing uses the Level-1B product and applies radiometric and geometric corrections (including orthorectification and spatial registration).

Finally, level-2A is an atmospheric correction applied on top of Level-1C products; further details in the technical notes.
For Sentinel-1 we have 3 levels as well, with internal differences in the processing stages due to the different nature of SAR acquisitions compared to MSI of Sentinel-2. In particular, we don't have anymore optical sensors, but we have devices measuring azimuth, elevation angle, polarization value, incidence angle.
Furthermore, in SAR data there is an additional phase, namely calibration/validation. In this context, it's important to firstly test SAR response to known input signals, and then validating it on other independent inputs.

2.2.2 Applications of AI for Earth Observation

Classification and Detection problems CNNs are very common in tasks such as image classification and detection of objects.

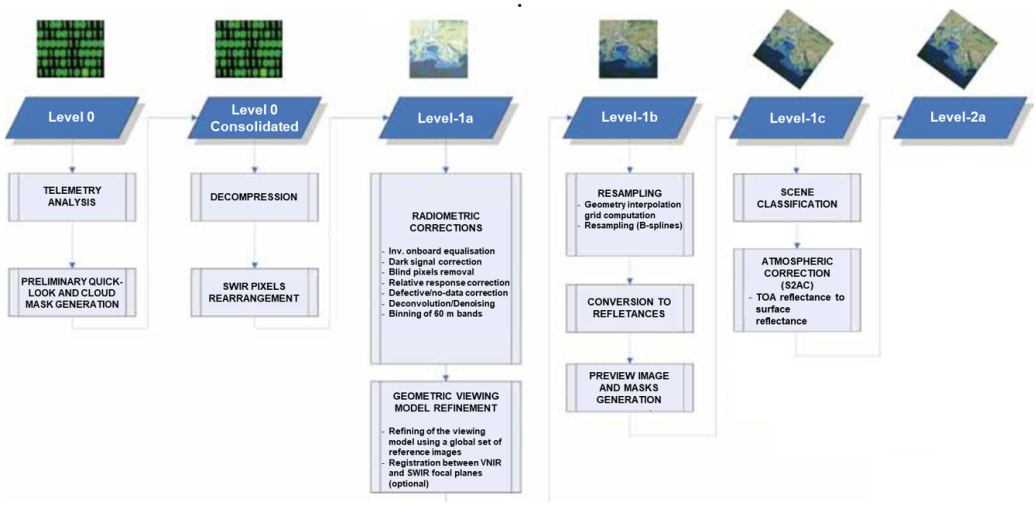


Figure 2.6. Sentinel-2 data levels, from Sentinel user guide

In literature, there are many approaches leveraging previous computer vision architectures to the remote sensing domain. As stated in [19], it's very difficult to perform transfer learning between classic computer vision data and EO datasets, therefore the majority of these works train from scratch a CNN.
There are several experimentations with the NN architecture, in particular concerning the backbone [20], the presence of attention layers [21] [22], the possibility of using it as a feature extractor serving as input for a second classifier, ensemble methods. Some common applications include land classification [23], vessel detection [24][25], wildfire identification [26], with the possibility of leveraging different EO sensors independently or by fusion approaches.

Biomass Estimation Biomass estimation is a common regression task applied to a typical EO domain. In fact, the goal is to to provide a detailed estimation of available flora (and fauna in some definitions) in order to extract valuable insights on the life cycles of the planet. Specifically, there are several correlations between the aboveground biomass size and the carbon cycle, thus this is an interesting factor to keep track of climate change.
In recent years, Artificial Neural Networks (ANNs) proved useful to overcome some traditional problems in classical AI methods, like simple regression, random forests and various esembles. This come from the "deep" nature of ANNs, which allows to capture and extract some valuable details from the images which would be ignored by shallow algorithms. Concerning the available data and sensors, a common approach is to combine Sentinel-2 and Landsat-8 images [27], to retain both the benefit of precision (Sentinel-2) and the benefit of retrocompatibility and estimation of Above Ground (ABG) biomass values.

Weather Forecasting This branch inherits all the neural network architectures connected with the time domain. Therefore, it has a large use of RNNs and relative variants [28], as well as GNNs ([29],[30]), which is the actual state of the art. As we will see in chapter 3, the generative models used for this thesis can be adapted as well for the time domain.
This branch has several approaches in various sub tasks, such as air quality prediction [31] and extreme events [32]. In this context, diffusion models may help to design

new and never-seen catastrofic scenarios, in particular running different simulations with different results due to their stochastic nature.

Change Detection and Semantic Segmentation Change detection is a very known problem in remote sensing; it's about spotting a change between two images captured in the same place at different times, ranging from couple of minutes, days or years. Semantic Segmentation, instead, is the task deputed to automatically segment an EO image to obtain a division into several classes of objects. Both these tasks are very important not only for the general EO community (see [19] for references), but also for our research problem. In fact, they can serve as downstream tasks for a possible generative model application.
In both domains we can play over many variables:

- Training and/or inference speed: we can try to optimize the models to reduce the computational workload required to reach SOTA performances, or even better we can try to reduce the inference time up to the point of bringing these models onboard.

- Network Architecture, because computer vision approaches have found a lot of variants over time, and an EO scientist must be careful in choosing the initial design of its network which is tailored to the problem; in fact, the time saved on NN architecture design will be lost in subsequent phases, trying to counterbalance with hyperparameter tuning some insite drawbacks of poorly designed NNs.
  Actually the most adopted architectures are the U-Net and the ViT. In particular, the U-Net has been successfully translated from the most general computer vision community to the segmentation of land portions, sea-land separation, urban deployment.
  The ViT is a sound choice in many circumstances because it captures both globally and locally the features of an image, thus skipping the necessary convolutional layers to obtain an acceptable receptive field in CNNs.

- Input Data. In this case, as before, an EO expert can decide wether band or sensor to use from the Sentinel products. For example, SAR is well-suited to perform change detection and semantic segmentation in cloudy scenarios. But they are not immediate to use, because of labeling difficulty, less available data than Sentinel-2 and a very challenging preprocessing pipeline.
  For this reason, many approaches try various sensor fusion strategies, such as Early fusion, Siamese neural networks and Late fusion approaches.
  If done properly, these techniques are able to retain the positive aspects of every sensor used in the pipeline. The fusion strategies may involve also more than two sensors. A comprehensive review of fusion approaches for remote sensing can be found in [33]

The abovementioned distiction can help us in deciding which CD method to choose for our downstream application. On the opposite direction, diffusion models parameters can be tuned to produce a personalized dataset for the end-user, depending on which method we currently uses.
CNNs have been widely used for change detection ([34]); In subsequent works,[35] ad [36], the authors propose a siamese approach to leverage the hyperspectral data for change detection.
Subsequent refinements focused on the various points we mentioned before: [37] experimented with attention, while [38] deployed a transformer architecture.

Other relevant applications are directed towards a fast computation model, TinyCD ([39]) and extensions to the 3D domain, as highlighted in [40].

Inpainting and Cloud Removal As in the previous paragraphs, we can model cloud removal with the same techniques used in more general computer vision problems, such as inpainting. Inpainting means removing a certain region of interest in the image and creating a new image portion in place of it.
In normal computer vision tasks, this finds application in several domains, such as masks weared by people, removing or changing a determined cloth on a person, removing an undesired object in an image.
In Earth Observation, it is straightforward to see the benefit of this technique on cloud removal. Actually, our region of interest is the portion of the image occupied by the cloud, and our goal is to replace the cloud with a coherent background, with smooth transitions along the borders and consistent with the ground-truth image.
Which techniques have been adopted to solve this problem?
Actually, inpainting in remote sensing is a key application for optical sensors, given their inability to receive signals from the land below the cloud.
As suggested in [41], cloud removal falls into some definite categories:

1. *Inpainting Only* in this way, an estimator leverages the information acquired from cloudy-free images, or comparing in training some pairs containing ground truth data and the cloudy one [42]. In this way, we can limit ourselves to few, selected bands to retrieve the necessary information, such as RGB.

2. *Data Fusion*: In this second case, the most common approach is to merge SAR data (Sentinel-1) with optical data (Sentinel-2), because radiometric data can see through the clouds. This can be done through simple composition ([43],[44]), or using some deep neural networks for more difficult scenarios [45] [46]

3. *Multitask approaches*: in this case, inpainting can be performed together with other tasks, such as cloud detection [47] [48].

4. *Multitemporal Approaches*: the first ones, as in the case of multispectral approaches, were bounded to a naive combination of cloudy images and corresponding cloud-free acquisitions at different times of the day, such as in [49], [50]; in subsequent approaches, this case, DNNs are used leveraging the time information to help the cloud segmentation and removal [51] [52].

In the deep learning context, CNNs were firstly adopted for cloud removal ([53], [54]). These early attemps shoewd the great improvements and potential given by deep neural networks, but still lacked specific adaptations to the required tasks. This is a generation problem, because we have to generate new data from missing part of the original ones; a discriminative architecture like CNN cannot do this job well.
For this reason, generative architectures have been widely used in the past, expecially concerning GANs ([55]), as showed in [56], [57],[58].
These approaches opened the road to more sofistications and refinements. In fact, the abundant availability of different data sources led to explore data fusion approaches and image translation strategies.
In the first case, a comprehensive work has been done by [59] with the realization of a complete and large dataset of cloud locations, followed by [60] which opened the road to different fusion strategies. [46] proposed a fusion strategy between SAR and optical images as well as [45].

Other approaches, instead, focuses on the image translation task between an optical image and SAR-like prodcucts.
[61] proposes one of the first approaches in this direction by leveraging a cycling-GAN for the translation part.
At the same time, [62] proposed a double simulation approach: we can simulate optical images both from a single SAR image and from multitemporal SAR-Optical images using a combination of CNN and cGAN.
Finally, [63] uses a two-stage approach with two different GANs: in the first one, they translate SAR images into optical images; with the second one, they remove the clouds.
[41] stress and underline the difficulty of training GANs for cloud removal, given their training instability. For this reason, they proposed a new approach, leveraging both a convLSTM for time-space blending and GANs to translate SAR into optical. In this way, they combine the translation task with the temporal information, in particular the convLSTM allows to feed the network with an already processed image, with an high percentage of removed clouds; in this way, the last step combines the two sections of the work as an input for the PLFM head (a U-Net), which finishes the task.
Concerning the spatiotemporal information, U-TILISE [64] provide a new formulation to leverage the time information.
The model consist of a modified U-Net architecture inspired to U-TAE [65], originally developed to translate timeseries into panoptic segmentation. It encapsulates three stages: 1) a hierarchical spatial encoder to project the image into the latent space 2) a temporal encoding in latent space to fill the missing parts using the temporal information and 3) a spatial decoder.
The interesting findings of their approach are 1) the absence of SAR data in the input information and 2) the direct recostruction of the whole time series instead of recostructing a single frame.
A final word must be spent on all the approaches trying to explain better the working principle behind many empiric results. In particular, UnCRtainTS [66] reconstruct images from cloudy timeseries providing both the image reconstruction and the corresponding aleatoric uncertainty; instead, in [67] there are consistent effort to explain the effects of clouds for cloud-based approaches such as classification and detection. This work focuses on two points:

1. *classification separability* it investigates the separability of cloudy and non-cloudy samples by inspecting the maximum probability, the entropy, the maximum information, the sum of logit values and the precision values; the idea behind this first concept is that a network *must be capable of express the level of confidence it has for a certain task*

2. *noisy samples*: in this case, the authors compute the saliency maps from the network outputs, leveraging Grad-CAM [68]. In particular, the so-generated saliency maps allow to spot the silent regions in the classifier receptive field.

Chapter 3

Diffusion Models

Diffusion models are the last generative model to be developed, and unsurprisingly they have a lot of room for improvement yet. The reason lies in the math formulation behind them. Except for their first formulation back in 2015, their real development started only in 2020; for this reason, they are still evolving, given the new theoretical formulations behind them, not to mention the countless new applications in which they are successfully used.
In this chapter, we will outline both their mathematical formulation and their applications in various fields.
They are both fundamental, because the first is necessary to provide theoretical background about the model, while the second one serves as an empirical evidence of the theoretical findings.
As we will see, the latter case can also show insights we cannot still easily extrapolate from the theory. We follow [69] in the general outline of the diffusion models framework and we highly suggest it as an introductory read to the topic. For a complete overview of the works, we suggest this website mantained from Valentin De Bortoli and James Thornton We will provide several subsections highlighting the most important works an line of thoughts applied on diffusion models. We will refer to some equations or findings as **milestones** to underline their importance as building blocks for the successive works.
In particular, we will introduce the subsequent milestones:

1. *Score Based Models*: the first strucured approach to leverage diffusion processes
2. *DDPM*: The first variational formulation of diffusion models
3. *DDIM*: The possibility of lowering the number of timesteps at inference time
4. *Diffusion Models Beat GANs on Image Generation*: SOTA achievement on image generation and introduction of **guidance**
5. *Latent Space*: diffusion models leveraging latent space rapresentation of input data
6. *Cold Diffusion*: rethinking the corruption process

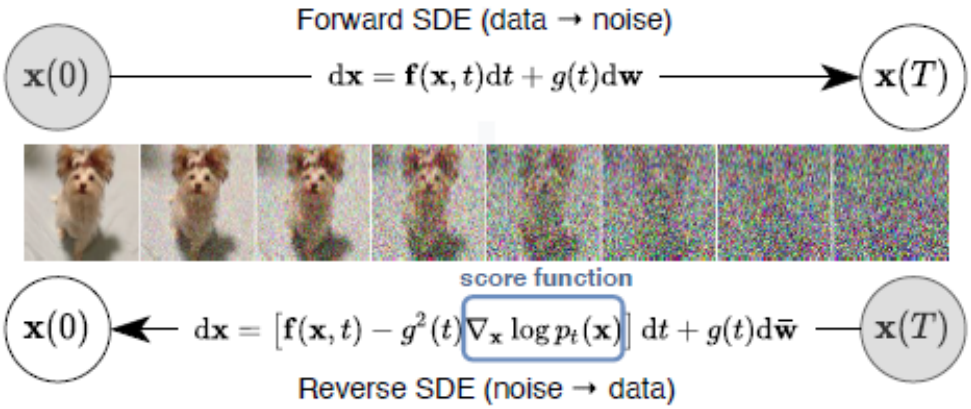


Figure 3.1. Image generation using SDEs, courtesy of [70]

3.1 Math Formulation

This section is important to explain why these models works and how do they obtain such results; in particular, it is necessary to investigate the (theoretical) limits in performance to provide a clear upper bound to reach in the field experiments.

3.1.1 Score Based Models

Diffusion models have been developed starting from a common stochastic differential equation, eq.1. This equation was first leveraged in [71], considered a seminal work in the field. In this work, the authors proposed for the first time to learn a data distribution by corrupting it following a diffusion process (given by an SDE) and then reverting it. This method, according to the authors, can be universally applied to any target data distribution because it exist a diffusion process for any smooth data manifold. From this moment, we will refer to this work as NSCN.
The general formulation of score-based models can be retrieved from [70]. In short, an image is progressively corrupted with noise following the stochastic differential equation (SDE) 3.1

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w} \tag{3.1}$$

Then, the image must be reverted back by computing the solutions to this SDE, as in 3.2

$$d\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g^2(t)\nabla_{\mathbf{x}} log\, p_t(\mathbf{x}]dt + g(t)d\bar{\mathbf{w}} \tag{3.2}$$

In fig. 3.1, we find an illustration of the proposed model. From [72], this result is assured to be a diffusion process as well. This means that it can be manipulated with the usual *Itô* formulation, allowing us to go back and forth with the diffusion process along an arbitrary number of timesteps. For further details on SDEs, we remind to a concise introduction [73] and to the base book for the field [74].
The seminal work from [75] deepen the previous work [71] by providing multiple alternatives in its usage. Firstly, they proposed score matching as a solution to learn the target data distribution. Given a data distribution $p(x)$ and an estimator $s(x)$, the process is learned by minimizing the following objective function:

$$\mathbb{E}_{p(x)}\left[tr(s_\theta(x)) + \frac{1}{2}\|(s_\theta(x)\|_2^2\right] \tag{3.3}$$

Where $s_\theta(x)$ approximates the gradient of the distribution p(x). There are some important considerations:
First, this model estimates the gradient of p instead of directly p. This allows us to skip the expensive learning of the original data to focus solely on his gradient Second, $s_\theta(x)$ is a variant of the original score matching network which avoids to compute higher order gradients Third, $tr(\cdot)$ is computationally expensive, so the authors propose these two shortcuts:

1. **Denoising score matching** This is the well known solution used also by subsequent works. In fact, instead of using s_θ to estimate the whole distribution, the data is firstly perturbed with a known function q, and then the score function estimates the new score $q(x) = \int(q(\tilde{x}|x)\dot{p}(x)$ with the new objective function:
$$\frac{1}{2}\mathbb{E}_{q(x)}\left[\|s_\theta(\tilde{x}) - \nabla_x log q(x)\|\right] \tag{3.4}$$

2. **Sliced score matching** In this case instead the authors approximate $tr(s_\theta(x))$ via random projections. The final objective becomes:
$$\mathbb{E}_{p_v(x)}\mathbb{E}_{p(x)}\left[v^T\nabla_x s_\theta(x)v\frac{1}{2}|(s_\theta(x)\|_2^2\right] \tag{3.5}$$
In this case, $v^T\nabla_x s_\theta(x)v$ is performed through forward mode auto differentiation, which is very precise, but it leads to large computational times. For this reason, the authors prefers to rely on item 1 for the rest of the work.

Challenges The authors identify two main challenges for these models:

1. **Manifold hypothesis** The Manifold hypothesis is fundamental in machine learning to reliably estimate data in a feasible amount of time. In particular, it states that the real information can be expressed as a low dimension manifold which is, however, embedded into an higher dimension distribution. This peculiarity penalizes a score matching framework, because $\nabla_x p(x)$ is undefined for low dimension x

2. **data scarcity** In this case, the score estimator cannot retrieve correctly the whole data distribution, but only the regions with enough samples. Moreover, in case of two-mode high density data distribution separated by low density regions, the model performance heavily depends on the inference scheme adopted for the test. The authors use exact sampling, Langevin dynamics and annealed Langevin dynamics

Given the aforementioned challenges, the paper introduces the first milestone of our diffusion framework. In fact, they propose to incorporate the noise information as an additional input to the score estimator, which becomes $s_{theta}(x, \sigma)$ for σ-based schedules; we rembember our initial blueprint, where the noise can be expressed also in terms of timesteps t or Signal-To-Noise Ratio SNR.
Moreover, given the difficulty in estimating the whole data distribution at once, the authors propose multiple noise levels to estimate back the data distribution one step at time. In this way, the injection of noise (and its estimation) will follow the rules of a Markov chain.

Further Improvements In [76] there are presented five techniques to attenuate the drawbacks of previous baselines, expecially concerning the impossibility of generating images at resolutions greater than 32x32. We resume them in the following bullet points:

1. *Choose σ_1 to be as large as the maximum Euclidean distance between all pairs of training data points.*, where σ_1 the first level of injected noise
2. *Choose $\sigma_{i}{}_{i=1}^{L}$ as a geometric progression with common ratio γ, such that $\Phi(\sqrt{2D}(\gamma-1)+3\gamma)-\Phi(\sqrt{2D}(\gamma-1)-3\gamma \approx 0.5$*
3. *Parameterize the NCSN with $s_\theta(x;\sigma)=s_\theta(x)/\sigma$, where s_θ is an unconditional score network.*
4. (selecting T and ϵ) *Choose T as large as allowed by a computing budget and then select an ϵ that makes Eq. (4) maximally close to 1.*
5. (EMA) *Apply exponential moving average to parameters when sampling.*

3.1.2 Variational Inference Formulation

Here we provide a new formulation for diffusion models, outlining DDPM work, its follow up DDIM and a key part of Variance Preserving techniques. The key difference with previous score-based works are 1) the re-formulation of the concept of diffusion noise estimation and 2) some improvements to the noise schedule.
The first point is critical, because the authors will show that the VLB objective for a time-discrete diffusion model, can be equivalent to multiscale denoising score matching, as highlighted in [77], therefore formulating diffusion models as a kind of VAE. Moreover, subsequent works [70] extend this findings to continuous time.

DDPM

In this scenario a major breaktrough came from DDPM (Denoising Diffusion Probabilistic Models) [78], which is our second milestone. In fact, this is the work on which we based our model architecture, and this is the work we refer in the math introduction to give an overview of what is a diffusion model.
The authors proposed two simple but important modifications to the NSCN framework:

1. Firstly, the noise levels and the associated variance schedules have much smaller values, thus leading to more stable training. We will talk about this later in the dedicated paragraph
2. Secondly, the authors propose to derive the estimator hyperparameters rigorously after the forward process; instead, in NSCN this parameters are hand-set.

The rest of the formulation is quite similar to previous works. The diffusion process can be divided into forward process (noise injection) $x \sim q(x)$ and denoising process (noise estimation) following $p_\theta(x)$.
The basic training principle of diffusion models is very simple, and we will distinguish between training and inference phase.

- **Training phase**: the input is incrementally corrupted along various steps $t = 1...T$ with gaussian noise $\epsilon_t \sim \mathcal{N}(0, \mathbf{I})$ of variance β_t.

- **Inference phase**: the model receives as input pure gaussian noise $\mathbf{x_T} \sim \mathcal{N}(0, \mathbf{I})$ with same dimensions of training input. The model progressively denoises $\mathbf{x_T}$ for every timestep, following equation 3.9.

In this work the authors adopt a noise value formulation, so the progressive noise injection is described in terms of β_t, with $t = 1...T$ and β as the variance of the injected noise.
Starting from x_0, the final sample is $\mathbf{x_T} \to \mathcal{N}(0, \mathbf{I})$ for $T \to \infty$. This is a very important result because the reverting procedure will assume to start from pure gaussian noise.
The forward diffusion equation is shown below:

$$q(\mathbf{x_t}|\mathbf{x_{t-1}}) = \mathcal{N}(\mathbf{x_t}, \sqrt{1-\beta_t}\mathbf{x_{t-1}}, \beta_t\mathbf{I}) \tag{3.6}$$

where q is the probability distribution obtained from the corruption process.
Moreover, we can derive directly $q(\mathbf{x_t}|\mathbf{x_0})$ as shown in (Ho?):

$$q(\mathbf{x_t}|\mathbf{x_0}) = \mathcal{N}(\mathbf{x_t}, \sqrt{\bar{\alpha}_t}\mathbf{x_0}, (1-\bar{\alpha}_t)\mathbf{I}) \tag{3.7}$$

where $\alpha_t = 1 - \beta_t$, and $\bar{\alpha}_t = \prod_{s=1}^{T} \alpha_s$ Thanks to the *reparameterization trick*, we can explicitly express $\mathbf{x_t}$ as a function of $\mathbf{x_0}$ and ϵ_t:

$$\mathbf{x_t} = \sqrt{\bar{\alpha}_t}\mathbf{x_0} + \sqrt{1-\bar{\alpha}_t}\epsilon_t \tag{3.8}$$

The noise is added following a β-schedule which adjust the amount of noise depending on timestep t, modifying the variance β.
Then, the model has to estimate the amount of injected noise into the input, so it has to estimate the probability q. This is called the *reverse diffusion process*. Shortly, we define an estimator $\epsilon_{\theta,t}$ which is,in our case, a U-net; the goal of reverse diffusion is to indirectly estimate the distribution q using an estimation distribution p
In formulas, we can sample the predicted $\mathbf{x_{t-1}}$ as shown in [78]:

$$\mathbf{x_{t-1}} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x_t} - \frac{\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x_t}, t)\right) \tag{3.9}$$

Training loss: from KL divergence to simpler formulations Firstly, for the forward process we have the sample mean $\mu_t = \frac{1}{\sqrt{(\alpha_t}}\left(\mathbf{x_t} - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\mathbf{t}\right)$ Secondly, using the variational lower bound as explained in [69], we obtain:

$$L_{VLB} = \mathbb{E}_{q(x_{0...T})}\left[log\frac{q(x_{1...T}|x_0)}{p_\theta(x_{0...T})}\right] \geq \mathbb{E}_{q(x_0)}logp_\theta(x_0) \tag{3.10}$$

This equation can be splitted also into several components based on the timestep t, as follows:

$L_{VLB} = L_T + \sum_{t=1}^{T-1} L_t + L_0$, where
$L_T = D_{KL}(q(x_T|x_0)||p_\theta(x_0)$
$L_t = D_{KL}(q(x_t|x_{t+1}, x_0)||p_\theta(x_t|x_{t+1})$
$L_0 = -logp_\theta(x_0|x_1)$
The simplification occurs taking into account 1) $D_{KL}(a|b) = \mathbb{E}log\frac{a}{b}$ and 2) using

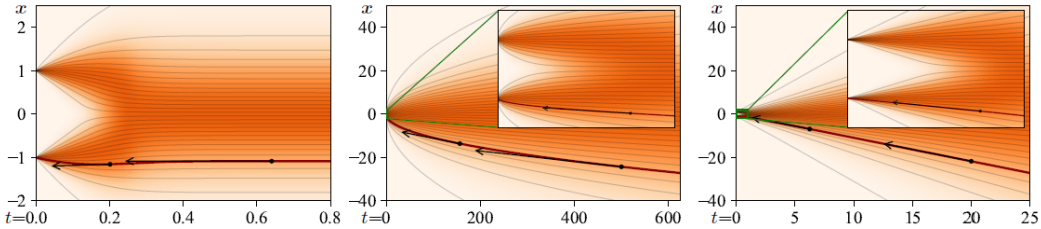


Figure 3.2. Left: VP ODE; middle: VE ODE; right: DDIM (from [79])

$\mathbf{x_{t-1}}$ as formulated into 3.9.
The final loss we obtain is (after some calculations):

$$L_t = \mathbb{E}_{x_0,\epsilon}\left[\frac{(1-\alpha_t)^2}{2\alpha_t(1-\bar{\alpha}_t)\|\Sigma_\theta\|_2^2}\|\|\epsilon_t - \epsilon_\theta(\mathbf{x_t}, t)\|\right] \tag{3.11}$$

where $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t$. This can be further simplified into

$$L_t^{simple} = \mathbb{E}\left[\|\epsilon_t - \epsilon_\theta\|\right] \tag{3.12}$$

Connection with NCSN models and langevyn dynamics Following the explanation of [69], we can explain the connection between NSCN models and DDPM formulaiton. In particular, how do we link the score network expression s_θ with the DDPM estimator ϵ_θ ?
We remember that $s_\theta(x_t, t) \approx \nabla_x log q(x_t)$.
We recall also that $\nabla_x log p(x_t) = \nabla x\left(-\frac{1}{2\sigma^2})(x-\mu)^2\right) = -\frac{(x-\mu)}{\sigma^2} = -\frac{\epsilon}{\sigma}$. Moreover, $q(x_t|x_0) \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, \sqrt{(1-\bar{\alpha}_t}I)$; this implies $\sigma = \sqrt{1-\bar{\alpha}_t}$ and therefore:
$s_\theta(x_t, t) \approx -\frac{\epsilon_\theta(x_t,t)}{\sqrt{1-\bar{\alpha}_t}}$

VP vs VE models

We briefly mention this small but important part of diffusion models formulation in the variational scenario. As explained in [70], diffusion models can be formulated equivalently in a Variance Exploding (VE) fashion or in a Variance Preserving formulation. Even though they are equivalent, is generally preferred to train a model in the VP setting, to avoid large error losses.
In formulas, the VE equation representing the corruption process is

$$d\mathbf{x} = \sqrt{\frac{d|\sigma^2(t)|}{dt}}d\mathbf{w} \tag{3.13}$$

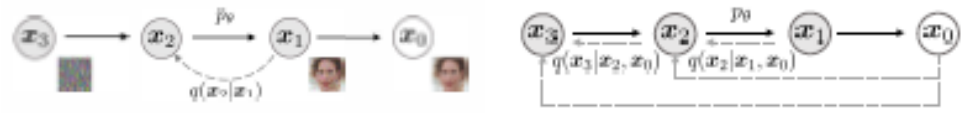
taken from the score-based formulation, where we can notice the variance value increment undefinitely as $t \to \infty$.
Instead, for DDPM the authors show that eq. 3.8, for $T \to \infty$, converges to the following SDE:

$$d\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}dt + \sqrt{\beta(t)}\mathbf{w} \tag{3.14}$$

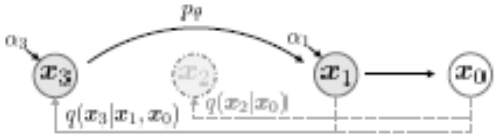
which leads to a fixed variance as $t \to \infty$. A nice visualization of the ODE dynamics in the different settings have been proposed by [79] in the context of explaining better the design of diffusion processes and the working principle of the equations regulating them.

DDIM

DDIM [80] can be considered the third milestone of this framework. The key working principle is simple: the model is trained using DDPM with the usual number of timesteps. Then, the inference schedule is sligthly modified to reduce the number of timesteps of the process. Therefore, this scheme allows for an usual training of diffusion models while allowing for flexibility in their inference phase. In fig 3.3a we can see an overview of the proposed method, which does not use anymore the markovian-chain formulation and moreover, talking in terms of equations, the SDE loses its stochastic nature becoming an Oedinary Differential Equaiton (ODE); in fig. 3.3b there is a sketch showing the new sampling procedure with a reduced number of timesteps. During generation, we sample from a subset $\tau_1, \tau_2 ... \tau_S$. Moreover, the noise



(a) New probability formulation



(b) Sampling procedure

schedule vary according to the newly selected timesteps, becoming $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$, with the new variance $\sigma_t = \eta \cdot \tilde{\beta}_t$, and $\eta \in (0, 1)$ a controlling factor for the stochasticity of the model.
Finally, we can write the final formula for the sampling probability to estimate:

$$q_\sigma(\mathbf{x}_{\tau_{i-1}}|\mathbf{x}_{\tau_t}, \mathbf{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t-\sigma_t^2}\frac{\mathbf{x}_{\tau_i} - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}}, \sigma_t^2\mathbf{I}) \tag{3.15}$$

Alongside DDIM, many efforts have been done to stretch the performance at inference time, with the release of various different samplers, namely PLMS [81], UniPC [82], DPM++ [83]

3.1.3 Cold Diffusion

This is another important milestone we mentioned in the introduction of this chapter. Cold Diffusion [84] redefines the paradigm for the corruption process of the image. The frequent question is: why do we need to apply gaussian noise to an image to make Diffusion Models work? Can we apply another type of noise? Can we redefine the concept of "noise"? Does it has to be a probability distribution?
The authors address all these questions in their work. In particular, they show that random noise and subsequent denoising process can be removed completely from the framework, leading to the most general concept of *degradation* and *restoration*, which they call operator $D(\mathbf{x}, t)$ and operator $R(\mathbf{x}, t)$.
In particular $D(\mathbf{x}, 0) = \mathbf{x}_0$ and $R(\mathbf{x}, t) \approx \mathbf{x}_0$. The newly defined loss becomes:

$$\min_\theta \mathbb{E}_{x\sim\mathcal{X}}\|R_\theta(D(\mathbf{x}, t), t) - \mathbf{x}\| \tag{3.16}$$

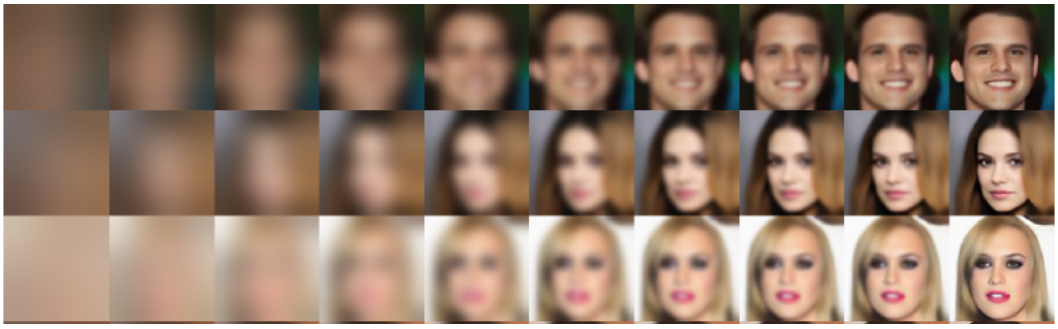


Figure 3.4. Blurring operator and deblurring

where $\mathcal{X}$ is the generic distribution from which we sample $\mathbf{x}$.
The authors propose several different types of "cold" diffusion (using a generic type of degradation), as opposed to the well-known diffusion process, remarked as "hot" (using noise), we leave a numbered list of them and an exaple image in fig. 3.4:

1. *Deblurring*: the degradation process apply a blurring operator to the images
2. *Inpaint*: the authors define a schedule of transforms that progressively grays-out pixels from the input image. They apply masks to the image to grey-out the pixels, with increasing blurring factor across timesteps
3. *Super-resolution*: at every timestep, the input image is downscale by a factor of 2 until arriving at 2×2 or 4×4 dimension.
4. *Snowification*: A non-traditional degradation approach, obtained by adding snow to the images

3.1.4 Countinuous time versus Discrete time

An interesting work from [85] challenges the usual formulation of diffusion models as discrete-time models. Their main modification, instead, is to propose a model where the corruption process happens at any time between two consecutive timesteps t and $t-1$, thus reformulating the framework as a Continuous Time Markov Chain (CTMC), as we can see from image 3.5 The authors provide then a reformulation

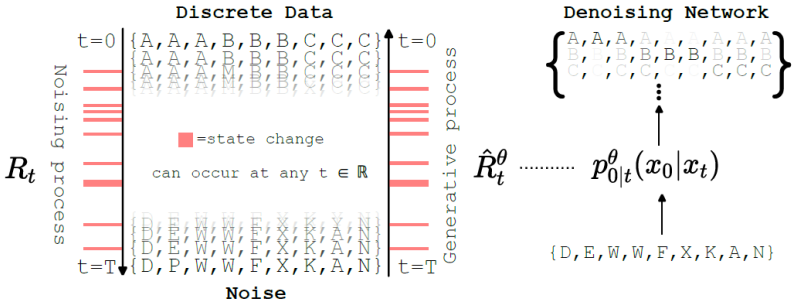


Figure 3.5. Method proposed in [85], where R_t is the rate of corruption events at time t and $\hat{R}_t^\theta$ is the corresponding estimated rate

for the ELBO loss, a new error bound and the reverse diffusion equations.

The authors notice that the quality of samples slightly improved, at the cost of multiple evaluations to be run and a worse final loss value

3.1.5 Convergence

We remember that diffusion models (and previously score-based models) construct a loss based on $\nabla_{\mathbf{x}} log\, p$, with p as target distribution. In [86] there are the first attempts to establish a convergence limit and an upper-bound for this quantity, but it was achieved only under restricting assumptions, such as the target distribution admitting a density w.r.t. the Lebesgue measure and under dissipativity conditions. The main problem is to trust empirically the loss from the visive results, without taking into account the term $\nabla_{\mathbf{x}} log\, p$ exploding for $t \to 0$, and the previous attemps did not release some important boundary condition, not to mention the *Manifold Hypothesis*. Therefore, [87] extended previous works by taking this hypothesis into account, and furthermore they provide also a *convergence rate* for the backward process of diffusion models.
Shortly, they propose a bound on the Wasserstein distance of order one between the target distribution and the predicted sample; moreover, they extend this result to the expected value of this metric, namely $\mathbf{W}_1(\mathcal{L}(\tilde{X}, \pi) \leq C$ and $\mathbb{E}(\mathbf{W}_1(\mathcal{L}(\tilde{X}, \pi)) \leq K$, with K, C constants, $\tilde{X}$ the predicted distribution and π the target one.

3.2 Architecture Improvements and direct applications

3.2.1 Guidance

Diffusion models have proved to be very efficient at learning a certain data distribution to generate new data from that.
But they still lacks flexibility. For this reason, [88] introduces a novel type of diffusion model, "guided" by an external input. This external guidance maybe be text, a class label, or another image.
This novel approach introduces a modification to the well-established reverse diffusion process.
Instead of computing just $\nabla_x logq(x_t) = \frac{\epsilon_\theta(x_t, t)}{\sqrt{1-\bar{\alpha}_t}}$, we have $\nabla_x logq(x_t, y) = logq(x_t) + \nabla_x logq(y|x_t)$.
Thus, we compute the first term on the right side and we collect the terms by $\frac{1}{\sqrt{1-\bar{\alpha}_t}}$, obtaining:

$$\nabla_x logq(x_t, y) = \frac{1}{\sqrt{1-\bar{\alpha}_t}} \left(\epsilon_\theta(x_t, t) - \frac{1}{\sqrt{1-\bar{\alpha}_t}} \nabla_{x_t} log f_\phi(y|x_t) \right) \tag{3.17}$$

From this formula, we redefine the estimator as

$$\bar{\epsilon}_\theta(x_t, t) = \epsilon_\theta(x_t, t) - \frac{1}{\sqrt{1-\bar{\alpha}_t}} \nabla_{x_t} log f_\phi(y|x_t) \tag{3.18}$$

This model can be further refined. In fact, it implies the usage of a second network $f_\phi(\cdot)$ as an external classifier, whose gradients "guide" the diffusion model.
Instead, [89] propose a model with classifier-free guidance. This means that we discard the external classifier and we use the network to guide itself.
Now we will briefly explain how.
Equation 3.17 can be redefined using ϵ_θ in place of f_ϕ; moreover, we choose to

compute $\nabla_x logp(y|x_t) = \nabla_x logp(x_t|y) - \nabla_x logp(x_t) = \frac{1}{\sqrt{1-\bar{\alpha}_t}}(\epsilon_\theta(x_t, t, y) - \epsilon_\theta(x_t, t))$. After some calculations, we obtain the new estimator:

$$\bar{\epsilon}_\theta(x_t, t, y) = (w+1)\epsilon_\theta(x_t, t, y) - w\epsilon_\theta(x_t, t) \tag{3.19}$$

where w is a weighting factor introduced to balance the contribution of the guidance. In particular, the network is trained using the labels y; in the original work ([89]) we can choose to drop 10% of the labels sampling the model unconditionally. After that, in inference, the network is used twice. The first time as $\epsilon_\theta(x_t, t)$, and the second time using the additional label y: $\epsilon_\theta(x_t, t, y)$. The final score is obtained as in eq. 3.19.
Important: We specify that the labels dropping does not mean that the model does not receive a label as input; instead, it receives a label which turns off all the weights of the network, like the zero element in a group. We will call it a EMPTY label $y = \emptyset$.
We make some examples. If the label is text, the corresponding EMPTY label is just an empty string "". If the label is a class, and the model receives class labels from 0 to 99, we can use 100 as EMPTY label. In the case of label as image, we can use an image full of zeros. This improvement implies two important consequences:

1. Better resource usage: We leverage the diffusion model both in unconditional and conditional scenario

2. Flexibility: we can tune the guidance scale factor w to test various guiding strategies during the inference phase

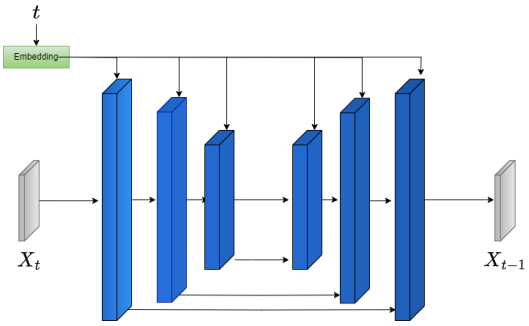
Moreover, these works are very important because they achieve, for the first time, better performances than previous State of the Art approaches, such as GANs, in image generation tasks. Finally, there are subsequent works trying to extend the guidance modalities and input types we can use in a diffusion model. A step towards this direction is made by [90]; in this work, the authors propose to use text-conditioned diffusion models with other input modalities, without retraining them. This introduces an important paradigm in the field, because these models can be turned into foundation models, capable of adapting to different scenarios.

3.2.2 Backbone choice

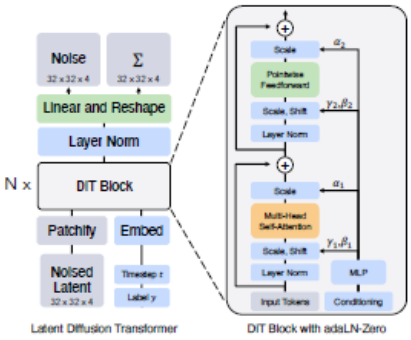
Here we provide a scheme of the different backbones architectures we may find in literature. This is important to show how much flexibility do we have with these models in terms of architecture design. Secondly, we can expose the main strengths and weaknesses of the various choices implemented in previous works. We list the following possibilities:

- U-Net backbone
- ViT backbone
- GAN-Diffusion combination

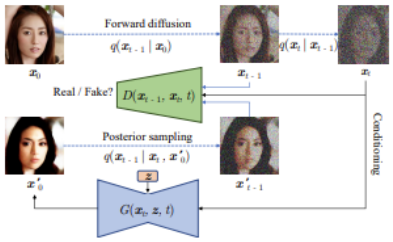
For item 1), we mention the commonly used U-Net architecture [92] Concerning item 2), [91] uses a transformer architecture to substitute the U-Net during the reverse diffusion process. This is an interesting approach, because Transformers, and in particular Vision Transformer (ViT) have proven to beat classic Convolutional



(a) U-Net backbone



(b) ViT backbone, from [91]



(c) Diffusion GAN hybrid approach, from [18]

networks when we need global spatial context. For this reason the original U-Net is usually slightly modified with the insertion of attention layers after the convolutional ones. Item 3) is very challenging and still unexplored. Actually, we will see later how diffusion models can model also videos. But it some special cases they may be unsufficient. In this scenario, we report the solution provided in [16], where the authors use the RNN output as an additional conditioning for the model at every timestep. At the best of our knowledge, nobody has tried to realize an RNN-Diffusion model in a joint fashion, maybe replacing the U-Net with an RNN, or inserting the diffusion process into the RNN model. Moreover, diffusion models suffer from long waiting time at inference, while at the same time VAE and GANs have different problems complementing each other, as we mentioned in section 2 which generated the generative Trilemma [18]. For this reason, [18] propose an hybrid approach to alleviate all the drawbacks of the three models.

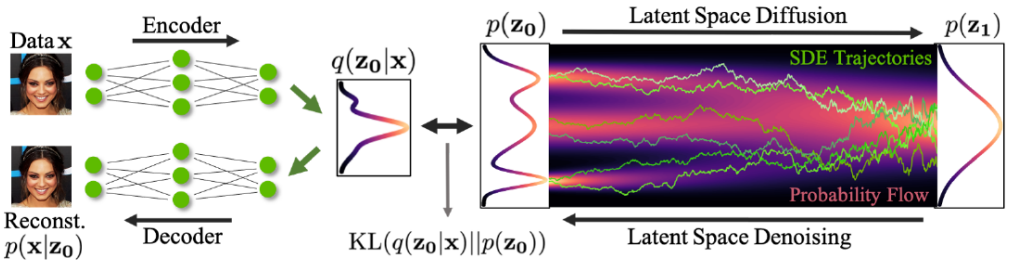


Figure 3.7. Scheme provided by [93]; the authors provide a first block to map the input data into the latent distribution, and then they show the diffusion process in terms of SDE trajectories; as we can see, the more we inject noise in an image, the more the distribution resemble the gaussian one, with the trajectories converging towards the highest value of the probability density

3.2.3 Diffusion models in latent space

We have previously provided to the reader some tools to guide a model towards the desired data distribution.
Sometimes, the process can be long and difficult due to an excessive size of the input data sample. For example, a 1024×1024 image can take several minutes to be rendered with 1000 timesteps using DDPM. Moreover, in training we may not need to start using the full input data, but we want to capture the few, important details for the task we perform.
A first exploration has been done by both [93] (see fig. 3.7) and [94].
In particular, [93] build upon score-based methods by projecting the input data into a latent space, using a VAE. More interestingly, the authors propose some key modifications:

1. A new loss taking into account both the VAE prediction and the Diffusion Model U-Net output

2. The provide a proof for an upper bound on the new defined loss, in particular concerning the cross-entropy term

3. Variance reduction: As we said in previous parts of this thesis, diffusion models in score-based formulation can be rewritten in VE or VP fashion. the authors design the SDE such that 1) it's Vp 2) They can sample via *Importance Sampling* and 3) Rewrite the VPSDE and a gemometric VPSDE by having $d\, log\sigma_t^2/dt$ constant for $t \in [0, 1]$

A similar concept is presented in [94]. The greatest difference compared to [93] lies in the latent space definition: in this case, the authors provide two improvements:

1. Generalization: they define a generic operator $\mathcal{D}$ to map the input into a subspace; then this is adapted to different type of mappings, such as downsampling operations for images. Moreover, the authors provide this as an integrable module into existing diffusion models approaches

2. Subspace selection: the authors present a new trick for subspace selection: they use *orthogonal fisher divergence*. This metric represents the error which would be introduced at timestep t. They state that this metric is independent on the previous subspace, therefore it can be considered as the only hyperparameter to tune during the selection process; moreover, this allows to plot simultaneously the corresponding value for every considered subspace.

Finally, we mention an important work of last year, Latent Diffusion [95] Latent Diffusion attach a VAE encoder on top of the Unet and a subsequent VAE decoder to map back the latent generated features to the original size. The main goal of this work is to compress the image by a factor f, usually $f \in \{4, 8\}$; in this way, the real diffusion process is applied only to a latent vector whose dimension is highly reduced, therefore speeding up the training and inference for high-resolution images, up to the Megapixel. As we see in fig. 3.8, the second major improvement within

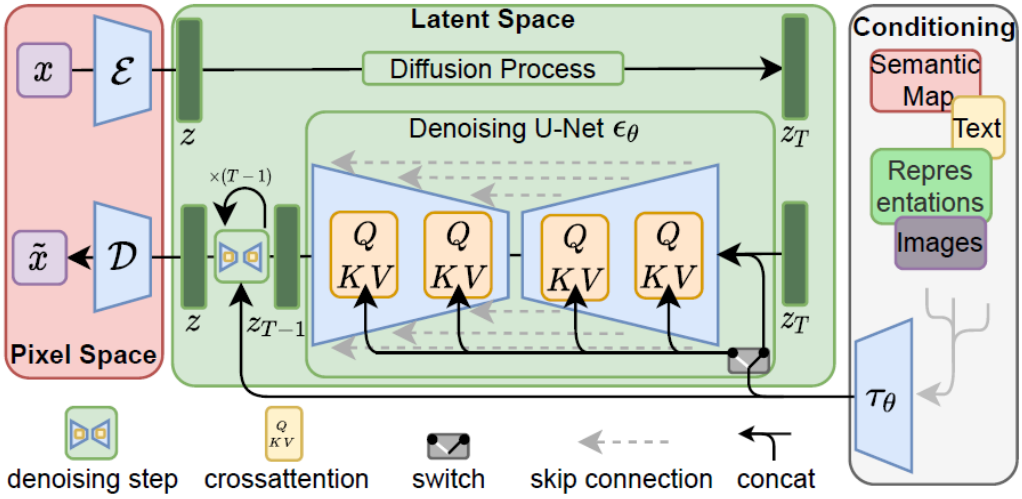


Figure 3.8. LDM approach, from [95]

this framework is the use of a universal module τ_θ which encodes the conditioning into the common latent space of the U-Net input data. Moreover, it can encapsulate the conditioning such that to feed its latent representation directly to the attention layers of the U-Net instead of the first block.
The conditioning can be of any type (text, image, class label) and LDM finds several important applications such as text-to-image generation, inpainting, super-resolution.

3.2.4 Video Diffusion Models

Diffusion models can be extended to a 4th dimension by leveraging time. There are several works in literature providing architecture changes for temporal sequences ([96], [97]). The main idea behind these models is to change the U-Net in order to accept 3D features instead of 2D, as shown in fig. 3.9

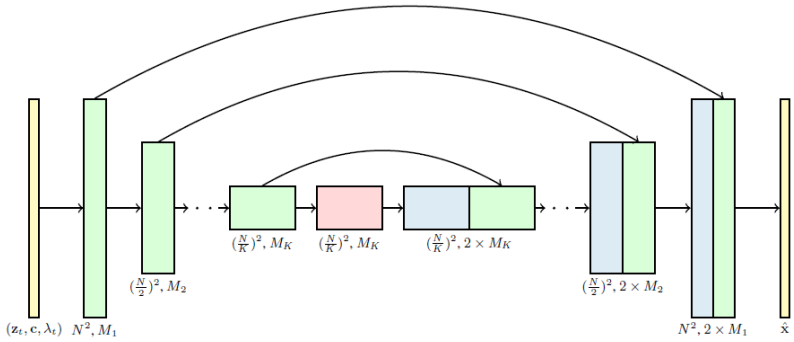


Figure 3.9. 3D U-Net Model

Is this the only modification we can do in order to adapt a diffusion model to time? Is it the best one?
Previous works, such as [98], may suggest to use cascaded models also in this case, and adapt them to the temporal framework. [96] provide the aforementioned concept of 3D-U-Net. But [97] builds on top of it and tries to leverage multiple frames at the same iteration, as shown in fig. 3.10. Furthermore, the U-Net is part of a larger cascaded pipeline, using inputs at different resolutions and different fps, as shown in fig. 3.11.

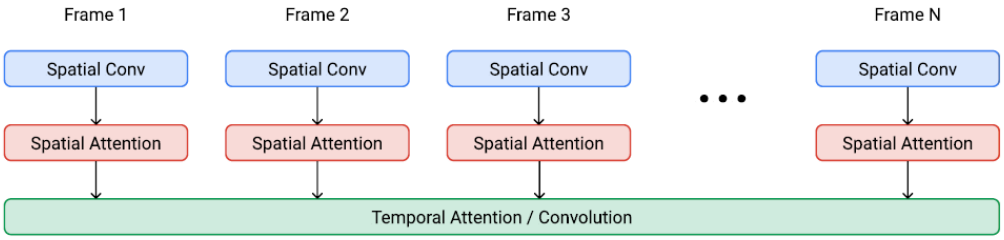


Figure 3.10. New U-Net architecture proposed in [97]

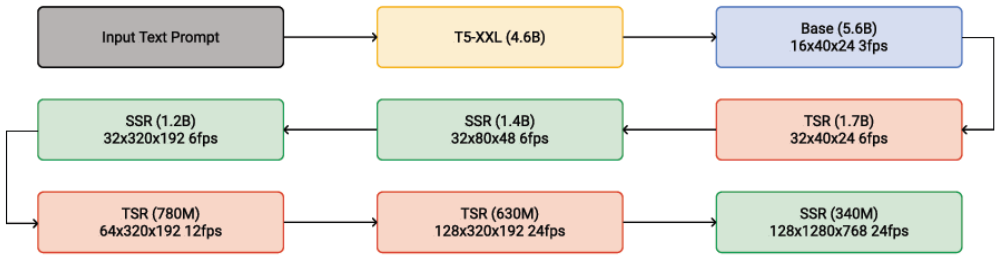


Figure 3.11. cascaded architecture in [97]

The key working principle of Video Diffusion Models is the **v**-prediction framework. It's just a reparameterization of the previous ϵ framework; in formula, $\mathbf{v}_t = \alpha_t \epsilon - \sigma_t \mathbf{x}$. According to the authors, this method allows to stabilize the training of image sequences, achieve faster convergence and better sample quality. Imagen introduces two other important updates, which can turn useful for following works:

The first one is the joint image-video training. The network is trained to generate a video sequence with the additional input of an independent image. The estimator becomes $\epsilon_\theta(z_t, t, c)$, where c is the conditioning image. This is passed to the U-Net as a 1-frame video, and concatenated to the rest of the video after skipping the temporal convolution resiudual blocks. This enhances a better transfer from images to videos The second one lies in some hidden improvements into the U-Net structure, as highlighted in [69] into three points:

- Shift model parameters from high resolution blocks to low resolution by adding more residual locks for the lower resolutions;
- Scale the skip connections by $1/\sqrt{2}$
- Reverse the order of downsampling (move it before convolutions) and upsampling operations (move it after convolution) in order to improve the speed of forward pass.

Subsequent improvements to this work are realized by Dreamix [99] and MagicVideo [100]. In the first case, the authors provide an editable framework to modify on-demand video sequences; in the second case, the authors extend Latent Diffusion Models for the time domain; a similar work has been done by [101] extending Stable Diffusion to inpaint specific guided type of clothes.
At the same time, [102] and [103] propose some further improvement in video quality generation. In the first case, the authors provide a method to control and generate highly customizable video sequences at high resolution and frames per second. The second work leverage RGBD images to synthetize a novel camera trajectory. The main novelty of this work lies in its focus on camera trajectory rather than in generating new framed images given the same camera pose.

3.2.5 3D Diffusion

Diffusion models can be extended to the 3D domain, as we show now and in section 3.3.2.
In particular, [104] [105] generates 3D images from text inputs, thus extending stable diffusion in its functionalities.
An interesting extension is to generate point clouds using diffusion models, such as Point-E [106] and [107]. We spend a few words to comment the first work. The authors do not train on point clouds to generate them (unconditionally or conditionally). The prefer to train on synthetic views generated with [108], and then training a diffusion model to output a point cloud based on this image, as shown in fig. 3.12 At this point, we may ask if we can merge the 3D representations with the

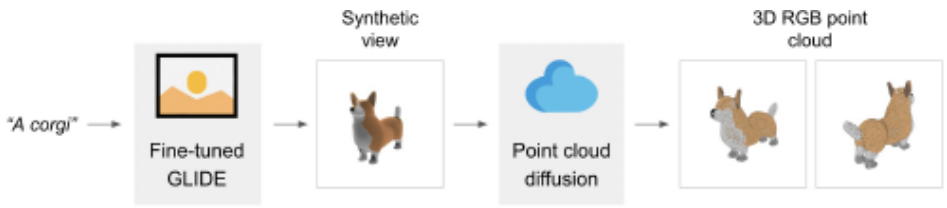


Figure 3.12. Point-E approach

time domain. The answer is yes, as in Text-to-4D by [109]; the authors provide a novel method to generate dynamic scenes from text inputs. Moreover, they provide

other modalities, such as Scene animation (Image to Video). We leave an overview of their approach in fig. 3.13

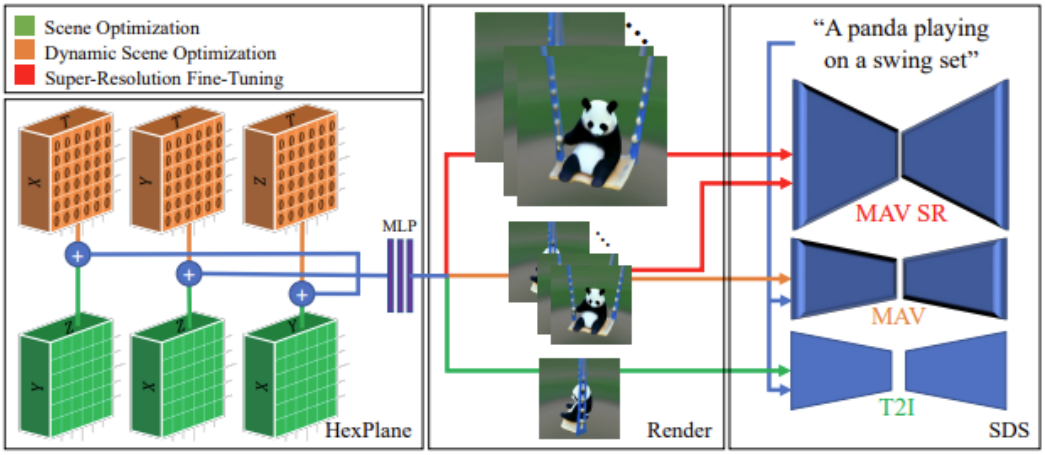


Figure 3.13. Text-To-4D model Architecture

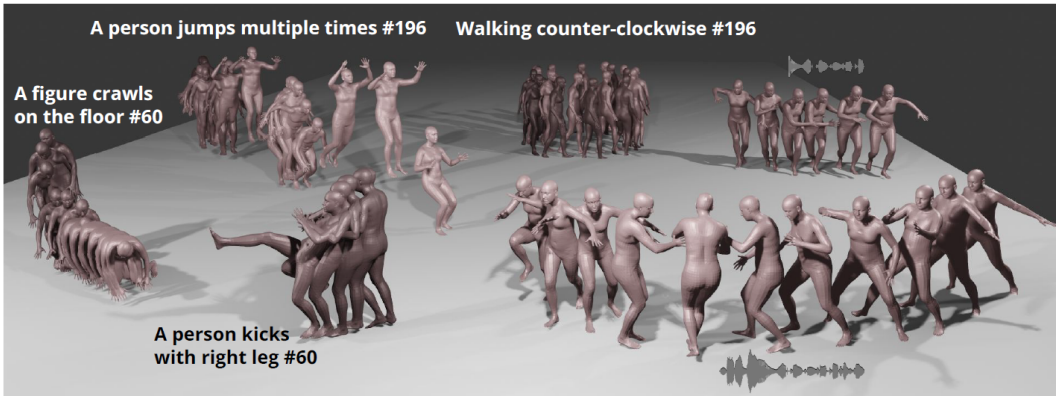


Figure 3.14. Character animation using diffusion models, image from [110]

3.3 Diffusion Models in other domains and applications

3.3.1 MOCAP

Motion Animation Capture is the field deputed to learn and describe human motion and gestures, using computer vision and computer graphics approaches. Diffusion models proved to adapt well to this application field.
the main focus for these models is:

1. (temporal domain) Generating new sequences of animation of a given character
2. (spatial domain) Generating new human shapes

Concerning the second item, we mention MoFusion [110], a framework to generate realistic and semantically aware sequences of human motion. In particular, the authors do not limit themselves to an unconditional generation task, but they leverage the flexibility of diffusion models by conditioning the task with music or text. We can see a sample of their work in fig. 3.14
The authors specify an interesting challenge for diffusion models: even though they are amazing to generate images conditionally, they sill lack explainability when the temporal component is taken into account, such as motion synthesis.

A new type of Loss This is an interesting finding from the authors, which can turn useful also for other domains, such as EO.
Motion synthesis is a difficult and particular task, which usually requires to retrain a diffusion model from scratch, because there pretrained models cannot leverage their ImageNet weights due to a large difference between the original dataset distribution and the target one.
The authors reformulate the loss $L_t = L_{da} + \lambda_k^t L_k$. L_{da} is the original diffusion loss term. But the interesting thing is the reasoning behind the L_k loss. This is necessary because the diffusion loss alone is not sufficient to generate realistic motion synthesis, leading to motion jitter, artifacts, illegal skeletons; this happens because l_{da} approximate well the initial data distribution, but fails to capture some specific task related details and constraints; this can be useful for EO tasks, given that diffusion models have not been tested on EO data.
The authors provide other two, small tricks to refine their model: first, they introduce a weighting factor λ_k to penalize motions which are too close to $t = T$, otherwise

it would end up in noisy motion generation; second, they leverage some physical constraints for skeleton coherence, bone asymmetry and motion ground supervision. The resulting loss is therefore $L_k = L_s + \lambda_a L_a + \lambda_m L_m$.
The overall architecture resembles the original one of diffusion models, as shown in fig. 3.15

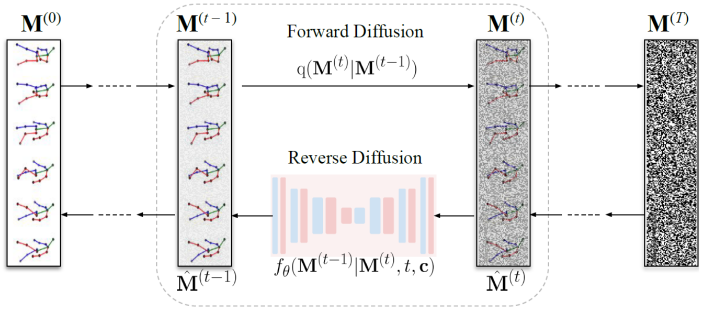
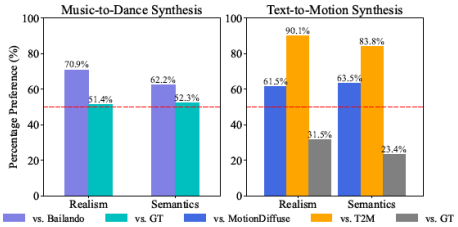
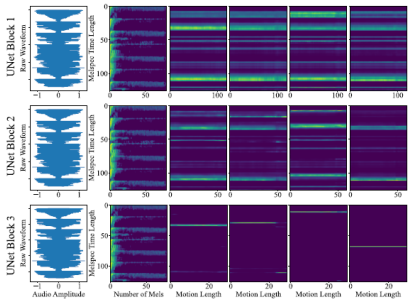


Figure 3.15. General Architecture of MoFusion approach

Useful insights from experiments The authors evaluate both the quality of generated images using FID and some task specific metrics, such as Beat Alignment Score and MultiModality score.
Interestingly, the authors provide a visualization of the U-Net weights (figure 3.16a), which can be a useful feature to shred light on the working principle of these models. Finally, the provide also chart to explain the task specific metrics (fig. 3.16b) [111]



(a) User studies evaluations on realism and semantics for Music-to-Dance synthesis and Text-to-Motion synthesis



(b) MoFusion visualization of crossmodal attention weights at different levels of the U-Net the U-Net

has a different approach. They show that we can use a diffusion model in latent

space, such as cite LDM, for a very difficult task such as human motion synthesis. Their methods proves to be an efficient alternative to standard diffusion models, and it opens up to the usage of latent diffusion models also in other domains, such as EO.
Other approaches referring to item 2 have been developed using diffusion models combined with established State of the art approaches.

There are other similar works, like [112], employ text driven approaches to generate sequences.
Finally, we refers to the work of [113]. In this paper, the authors are capable of generating meaningful and long sequences (minutes) conditioning them to 10-seconds short videos in training. This confirms that diffusion models can be trained to generate longer sequences even with short training videos, as already shown in [97]), [96].
We leave a list of approaches in this area to this website.

3.3.2 3D Reconstruction

Reconstruction of 3D environments is a well known problem in computer vision, with applications in architecture, autonomous navigation, game engines and many others.
The techniques we are going to present find a straightforward application in Earth Observation too, expecially concerning digital twins; there are other branches developing 3D reconstruction models for remote sensing, such as 3d models of clouds or building a 3D model for the Sun.
All these solutions use the well established baseline of NERF (NEural Radiance Fields) [114] approaches to reconstruct 3D environments from 2D images.
Diffusion models have been capable of integrating into other stable architectures, and this type of task (generating 3D scenes) is very favourable to their generative nature.

Challenges The 3D world present some novel challenges, as outlined in previous works on 3D diffusion models (cite cite).
It is straightforward to apply a 3D model given 3D data as supervision. But the real challenge is to:

1. leverage 2D images instead of 3D
2. Reduce the computational cost, which is already quite expensive for 2D diffusion models
3. guidance: we have to clarify which modalities to use (text, speech, other images...), and which domain to use (point clouds, grids, voxels...)

One of the first works to address these issues is [115]. In this work, the authors generate unconditionally 3D shapes, and they provide also a method to inpaint based on a 3D structure.
IC3D [116], instead, focuses on conditioning the model using a 2D image and they also use DDPM to generalize voxel-based data, thus introducing a new data representation which was avoided by the previous literature.
A similar solution has been addressed by HoloDiffusion for the grid domain, with the additional challenge of lowering the grid resolution and optimizing the memory usage.
DiffusionNerf and NerfDiff ([117], [118]) study slightly different problems: the first work aims at generating novel RGBD scenes given RGB based data; the second one synthetize scenes using both NERF renderings and Diffusion Model generation output. The key component is their focus on 3D view conditioning, which is in contrast with the literature trend of leveraging 2d data only. The same line of research has been adopted by DiffRF [119].
MeshDiffusion [120] introduces a relevant novelty, working mainly with mesh representations. Their work is essentially concurrent with Get3D, with the difference of using RGBD images as conditioning instead of SDF (signed distance field) and diffusion models as main network instead of GANs.
Finally, we mention two important improvements in the field: LION [121] and Plotting behind the scenes [122].
The first one introduces for the first time Latent Diffusion Models into the NERF architecture. The second one, instead, leverages both NERF and Diffusion Models to perform interactive view synthesis.

Noticeable architectures improvements and applications In this paragraph we will briefly show some relevant ideas and architectures which can be leveraged in future EO applications as well.
We will follow this outline:

- IC3D: how to use Diffusion Models and NERF using voxels
- MeshDiffuison: Applying a diffusion process with a mesh as input
- LION: Latent Diffusion into NERF
- Plotting behind the scenes: leveraging diffusion models for flexible video editing

Concerning IC3D, the interesting idea behind this approach is to extract two types of conditioning from the same image (see fig. 3.17):
First, the image is projected into the latent space of the network input.

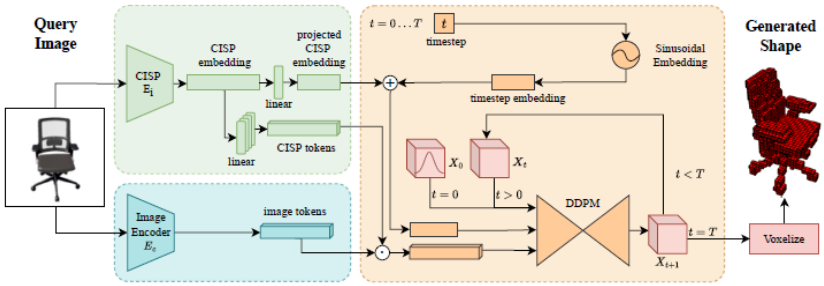


Figure 3.17. General architecture of IC3D [116]

Second, the image is decomposed in parallel into CISP embeddings, which will be concatenated to noise embeddings; moreover, the CISP (Contrastive-Image-Shape-Pretraining) projection of the image is further forwarded to the attention layers of the neural network as additional tokens.
This contrastive learning framework helps the authors in finding relevant information, leveraging the (image,shape) pairs in training.
Moreover, this paper is extremely useful to explain the relevance of the embeddings, as show in this scatter plot (fig. 3.18)

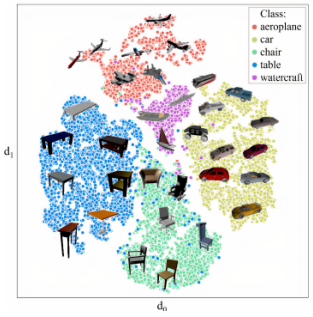


Figure 3.18. Projection into low dimension space of the generated samples

Now we give an overview of MeshDiffusion [120]. In this case, we want to show how a different image representation can still be learned using diffusion models.
Usually, they are used to learn 2D images data distributions, but in this case they

are extended to a totally different domain. This is the same approach we will see in GNN.
In particular, the diffusion model does not work with raw mesh; instead, the mesh is transformed into a cubic grid, such that the model can learn to generate meaningful cubic grids and the they are mapped back into a mesh.
From fig. 3.19, we can see the proposed approach, which is the equivalent of LDM, where the given image embedding is obtained from a VAE, while in this case we have a conversion from image mesh to cubic grid. LION is a parallel definition of

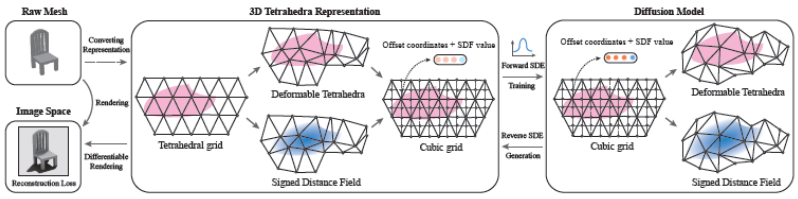


Figure 3.19. MeshDiffusion model overview, from [120]

the previous approach. The key is to redefine the latent space according to the given domain.
In this case, a latent embedding is intepreted as a shape outline, with fewer points than the original ones, but still sufficient to resemble the target shape.
This comes from an analogy with human intuition, where a person can deduce the object type by just a stylized shape, without too much details. See fig. 3.20 for an overview of the method

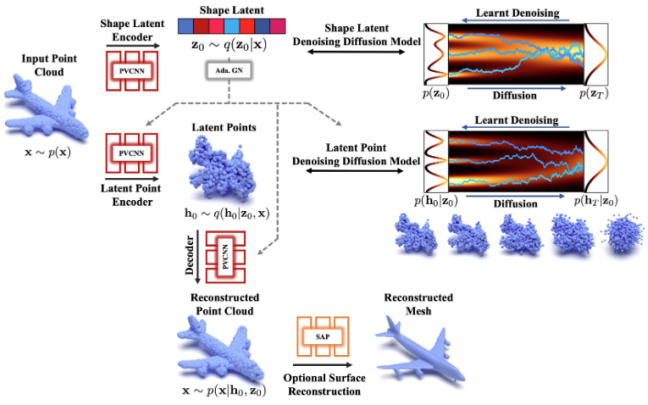


Figure 3.20. LION model outline, from [121]

Plotting behind the scenes is not only a fusion between diffusion models and 3D reconstruction. Instead, it leverages this new architecture to generate new sequences (thanks to diffusion models) starting from a given video, and the rendering pipeline is used to map the generated image into a 3D model.
This work make also use of guidance flexibility for video generation. In fact, it has been trained to change some timestamps of a video replacing them with a text-conditioned video sequence.
This novel paradigm has been defined as learnable environment from the authors, because it resembles the replay dynamic of a videogame. We leave a scheme of this

method in fig. 3.21

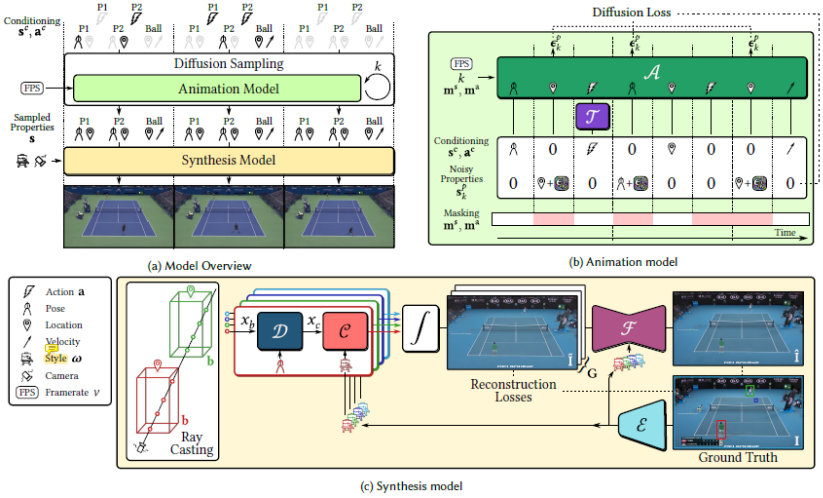


Figure 3.21. Plotting behind the scenes: the architecture is composed by a diffusion model and a NERF implementation

3.3.3 Graph Neural Networks

Graph Neural Networks (GNNs) [123] are a special type of neural networks working on graphs to learn the data distribution.
Their structure has been designed to deal with graph data representation, a problem which introduces new challenges compared to classical grid domain data, such as images.
For this reason, GNNs have founds several applications in brain cells analysis, protein folding and social network analysis.
Diffusion models can be tested in such environment to help the generation of graph data. In particular, given the generative nature of these models, they may turn helpful to generate new graphs, and therefore to help fields such as protein synthesis and drug discovery.
But this translation is not immediate. In fact, diffusion models must be tested over a different domain, which is the graph.
Until this moment, these models have been successfully applied to grids (images).
This statement bring us to the two following questions:

- Why should we be interested in GNN application of diffusion models? Because it study the flexibility and adaptability of this model to a new data representation, and this can be leverage in EO scenarios whith non trivial data. Second, some GNN approaches reformulate some key concepts of diffusion models, contributing to their mathematical development and therefore giving a structural update to the whole community behind diffusion models.

- Why this adaptation is not immediate? Because in a graph, unlike a grid, what changes is not only the information (like the RGB values of an image), but also its disposition in the domain space (like nodes changing relationships, positions, importance...).
  Moreover, as reported in cite digress, graphs "*have varying sizes, permutation equivariance properties, and to this date no known tractable universal approximator*"

The most basic application is to use diffusion models to generate graphs.
In these early works, the focus is solely on unconditional graph generation, with the main goal of testing the effectiveness of DMs on this domain too. Some works have separate sections with marginal experiments on conditioning too

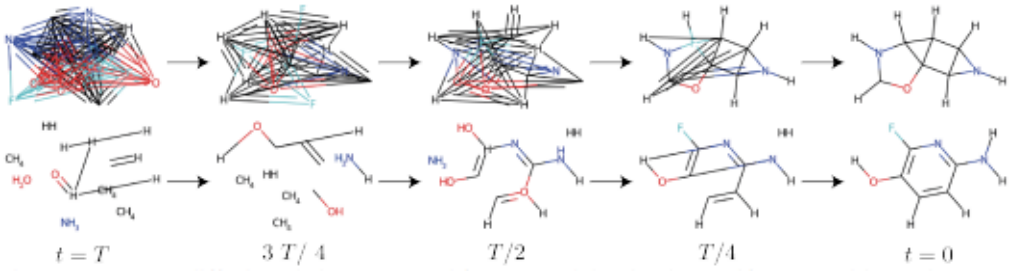


Figure 3.22. Digress denoising routine

DiGress: generating graphs with diffusion models DiGress [124] is one of the first works to address the aforementioned challenges.
As we can see in fig. 3.23, the authors have to define the equivalent of $\mathbf{x}_T$ for graphs

as the initial noise; in this case, we have G_T, which is a graph with island and three maximum connected nodes per island. As in the grid case, where the noise is added independently from the pixel location, in this case it is independent of the node-edges pairs. The diffusion model parameterization is not that much different from the standard one.
The main changes are in the loss function, formulated as a cross-entropy (CE) with the following formula:
Having a graph $G = (X, E)$, and transition probabilities $p^G = (P^X, P^E)$, the loss term is just the sum of crossentropies related to each feature (node-edge pair):

$$L(p^G, G) = \sum_{1 \le i \le n} CE(x_i, p_i^X) + \lambda \sum_{1 \le i \le n, 1 \le j \le n} CE(e_{ij}, p_i^E) \tag{3.20}$$

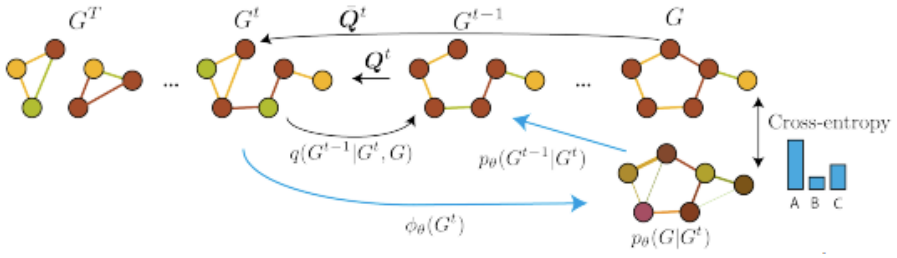


Figure 3.23. Digress architecture, from [124]

The network takes therefore a noisy graph G_t as input and produces the tensors X_{t-1}, E_{t-1}, and it is designed as a graph transformer neural network. In fig. 3.22 we can see the whole denoising process for a graph.

The authors need to define some properties for their method, because it is straightforward to check the grid invariance to shifts in convolutions (CNNs), but it's less straightforward to do so in GNNs with generic graph domains.
In particular, they claim the following three properties:

1. *(Equivariance). Digress is permutation invariant*
2. *(Invariant loss) The loss specified in eq. 3.20 is permutation invariant*
3. *(Exchangeability) DiGress yields exchangeable distributions, i.e., it generates graphs with node features X and adjacency matrix A that satisfy $P(X, A) = P(\pi^T X, \pi^T A \pi)$ for any permutation π*

Finally, we mention other two important improvements:
Firstly, the common uniform formulation of variance noise $\mathbf{Q_t} = \alpha_t \mathbf{I} + (1-\alpha_t)(\mathbf{1}_d \mathbf{1}'_d)/d$ is uniform in common grid-based diffusion models, but it leads to a limiting uniform distribution for node-edges pairs, which does not resemble their true disposition.
Therefore, they need a new formulation based on the graph node mean $\mathbf{m_X}$ and edge mean $\mathbf{m_E}$, obtaining the new formulas: $\mathbf{Q}^t_X = \alpha_t \mathbf{I} + \beta_t \mathbf{1}_a \mathbf{m_X}$ and $\mathbf{Q}^t_E = \alpha_t \mathbf{I} + \beta_t \mathbf{1}_b \mathbf{m_E}$

After these works, the subsequent formulations focused on protein folding-unfolding and drug discovery ([125],[126],[127],[128]).

DiffDock, molecule docking using diffusion models In this work the authors draw the outline for molecular docking, redefining the whole process in a generative fashion using diffusion models.
The interesting takeaways from this work are 1) the motivations to redefine such a task using diffusion models and 2) the changes made from the authors to the original diffusion models formulation.

The motivation behind this change of paradigm lies in two problems: *data ligand uncertainity* and *epistemic uncertainity*.
The first one is due to the ligand binding process, which can happen with multiple poses.
the second one is related to the disparity between the complexity of data and the relatively naive architectures adopted by the models.
Previous regression based methods were unable to capture all the modes in protein docking, let alone to identify the target one.
Diffusion models, instead, can generate multiple modes from the same protein thanks to their inner generative nature.

The inner modifications to diffusion models formulation is highly dependent to the target domain, and it needs to be functional to it. However, these can be interpreted and redefined for future EO related tasks with difficult data distributions and physical constraints; this could be the case for SAR or Hyperspectral data.
With a certain grade of aproximation, we can consider this similar to the working principle of latent diffusion, where the latent space is a totally different manifold in this case. We provide the method outline in fig. 3.24 For this reason, the authors

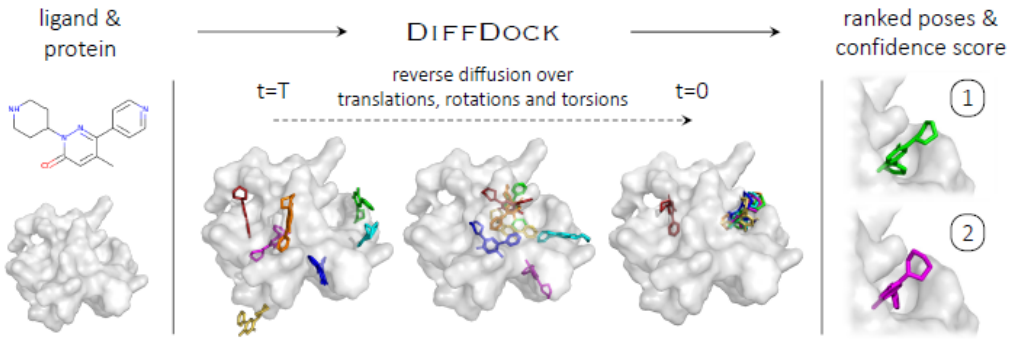


Figure 3.24. DiffDock method overview, image courtesy of [127]

define a new space, called product space $\mathcal{P} = \mathcal{T}^3 \times SO(3) \times SO(2)^m$, resulting in a compositon of three manifolds; this space is necessary to take into account the physical constraints related to the problem of ligand bindings, expecially concerning rotations and translations.
Another interesting result. according to rodola et al, is the possibility to sample independently each of the three manifolds contributing to $\mathcal{P}$

3.3.4 Audio models

Concerning audio models and music driven method, there are several works who contributed to develop the field.
We mentioned already MoFusion [110], who realized music guided generation as one

of the two subtasks of their work.
We can distinguish into two different categories, based on audio usage: unconditional audio generation or audio-based conditioning

Audio generation There are many works deputed to generate reliable audio effects, in particular reproducing human voice, or generating new music ([129],[130],[131],[132]). The key component is the encoding part of the audio signal, which has to be mapped into the U-Net backbone use to perform noise estimation.
In particular, both [133] and [132] diffusion propose a multimodal approach which separates in parallel the generated modalities, as shown in fig. the first one focuses

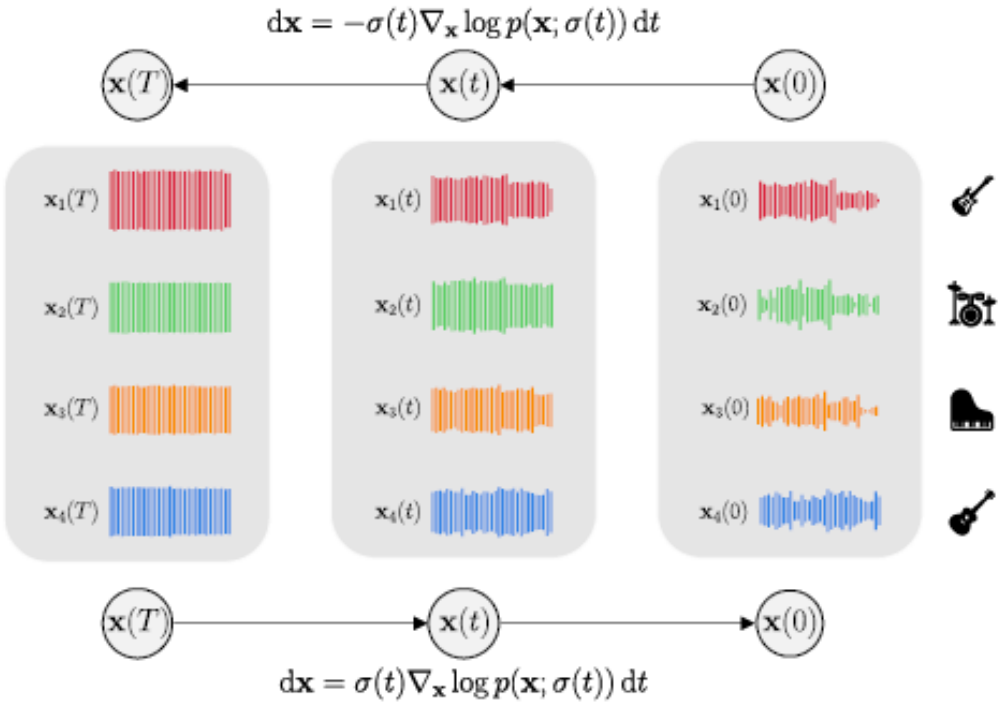


Figure 3.25. Multisource audio work from [132]

on joint audio-video generation, while the second work diffusion focuses solely on music generation, trying to separate different type of instruments audio sources. Other useful references are this implementation from weight and biases as well as this codebase, which is a master thesis work.

Audio conditioning This task will have several followups in the future because it's a relatively unexplored area.
In particular, [134] is not only a conditioned generator, but audio is a second conditioning step, which helps performing the real conditioning task. In this work, the task is video editing driven by audio. Another interesting paper in the area is [135]

Chapter 4

Model Formulation

In this chapter, we are going to present the architecture and mathematical development of our proposed solution.
In particular, diffusion models can be divided into two parts:
In the first one we will provide all the mathematical details relative to the diffusion process, namely the forward process, the reverse process and the relative scheduler type; every mentioned part embrace a set of different parameters which must be taken into account in the training phase. Moreover, we will show how to modify the reverse process in order to adapt it to our specific task.
In the second part we will deep dive into the estimator architecture, motivating its design and its choice.

4.1 Diffusion Model Framework

There are some important variables and parameters to bear in mind when dealing with diffusion models:

- *Timesteps* The number of iterations in our progressive noise injection $t = 0, 1...T$
- *Variance schedule* β_t, which represents the amount of injected noise and the corresponding corrupted image tend to the following distribution, as stated in eq. 3.6. As highlighted in chapter 3, we can have also different formulations based on Signal-To-Noise Ratio (SNR). It can be choosen as linearly decreasing, cosine-decreasing or other formulations. Moreover, it can be choosen to be set as an additional learnable parameter by the network.
- *Scheduler*: the previous parameters can be fed as input of a scheduler type, in particular concerning the β-schedule type
- ϵ-*formulation vs* x_0-*formulation*: this type of formulation can be found in some works implementing diffusion models, and it renders the term x_{t-1} as a direct calculation from ϵ or from x_0. The two definitions are almost equivalents because ϵ is, in turn, directly dependent on x_0. The only difference is that in the first case the estimator ϵ_θ compute the estimated noise ϵ from x_0, while in the second case it maps back an image into the next-step image x_t.
- *Loss term* this loss can be formulated as an MSE loss (l1 or l2), as shown in eq. 3.11, which can be simplified into:

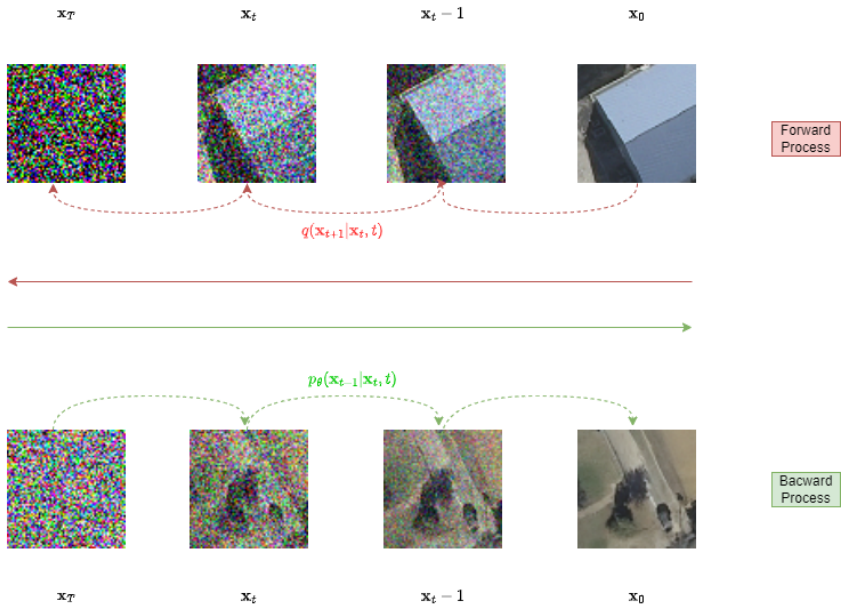
$$L_t^{simple} = \mathbb{E}\left[\|\epsilon_t - \epsilon_\theta\|\right]$$

- *Guidance type*: Guidance can be realized in several ways combining different modalities, such as text, class labels, images. Usually we redefine the estimator to take into account the additional label y obtaining an estimator $\epsilon_\theta(\mathbf{x}_t, t, y)$

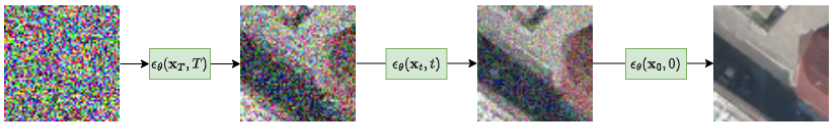
As we showed in chapter 3, we model the progressive noise injection as $x_t \sim q(x_{t-1}, t)$. This formulation creates one state depending only on the previous one, so can be seen as a Markov Chain.
At the same time, we model the progressive noise reversion with a probability p such that $x_{t-1} \sim p(x_t, t)$. We have to model a new probability p because q cannot be solved in closed form and reversed. For this reason, we need a noise estimator $\epsilon_\theta(\mathbf{x}_t, t)$ predicting the noise injected at timestep t, following eq. 3.9 which we report below:

$$\mathbf{x_{t-1}} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x_t} - \frac{\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\mathbf{x_t}, t) \right)$$

In image 4.1a and 4.1b, we report a general working scheme for our diffusion approach. We will talk more in the detail about the conditioning part in the neural network



(a) Training scheme for diffusion models



(b) Inference scheme for diffusion models

implementation, because it's the space where most commonly we apply conditioning operations. For the scheduler part, we will rely on DDPM formulation with $T = 1000$ timesteps.

4.2 Neural Network Architecture

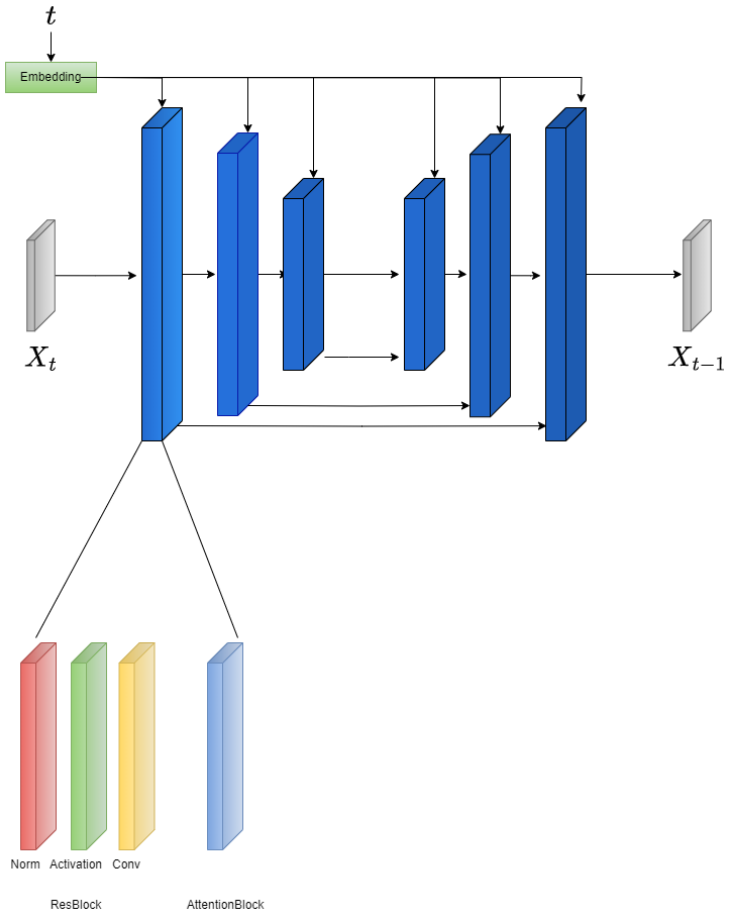


Figure 4.2. Unet architecture used in this work. We can see how the base model takes the input image and the corresponding noise information

Unet [92] is a convolution-based neural network architecture with an encoder block and a decoder block, often equipped with skip-connections between the two blocks to enhance gradient passing through the Unet. We will use this architecture as the estimator ϵ_θ for our reverse diffusion process, having the goal of denoising the image at timestep t estimating the $t-1$ level of noise. We borrow the base model used in [136], using a res-net backbone. As we see in fig. 4.2, the network has several layers, each one of them encapsulating:

- A time-steps embeddings block with an activation function (SiLu) and a linear mapping to the resblock channel dimension
- A Res-block with a normalization layer (groupnorm), an activation function (SiLu) and a convolutional kernel with a pooling operator
- An attention block with a parameter number of heads to be tuned

We will provide the details of everyone of them in the single experiments we will show. We will use mainly a group Normalization layer, together with SiLu activation

functions. The attention layer is optional, and can be used as a parameter for subsequent ablation studies.
Concerning inpainting, we can model the conditioning $y = x_c$ obtaining the above-mentioned estimator $\epsilon_\theta(\mathbf{x}_t, t, y)$. We can further define the type of conditioning in two ways: *concatenation-based* and *summation-based*.
The first one is the most widely adopted approach for image-conditioned DMs [136][95] and takes a concatenation on the image and the input along the channel dimension; therefore, we can define a new input $x_t^{concat} = x_t \oplus x_c$, with $x^{concat} \in \mathbb{R}^{(C1+C2)\times W\times H}$, and our network $\epsilon_\theta(\mathbf{x^{concat}}_t, t)$ architecture can be redefined as in fig. 4.3

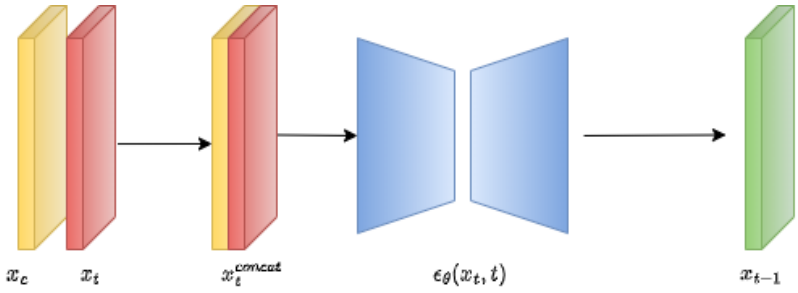


Figure 4.3. Channel-conditioning approach

The second one has been proposed by [137] and realizes concatenation as a sum between images. This type of conditioning can be advantageous because it does not need to retrain an unconditional model from scratch adding channels to support the concatenation, and can be leveraged inside every diffusion reverse step to enhance the generation of the image.
Therefore, we can define a new $x_t^{sum} = x_t * mask + x_c * (1 - mask)$ which takes the estimator output x_t. At first stage we have $\mathbf{x}_{T-1} = \frac{1}{\sqrt{\alpha_T}} \left(\mathbf{x}_T - \frac{\alpha_T}{\sqrt{1-\bar{\alpha}_T}} \epsilon_\theta(\mathbf{x}_T, T) \right)$; then, we map this estimation into x_{T-1}^{sum}. From this stage on, the iterative process will always bring back to the estimator the summed conditioning x_t^{sum} through the iterative assignment $x_{t-1} = x_t^{sum}$ We show the new scheme $\epsilon_\theta(\mathbf{x}_t^{sum}, t)$ in action in fig. 4.4

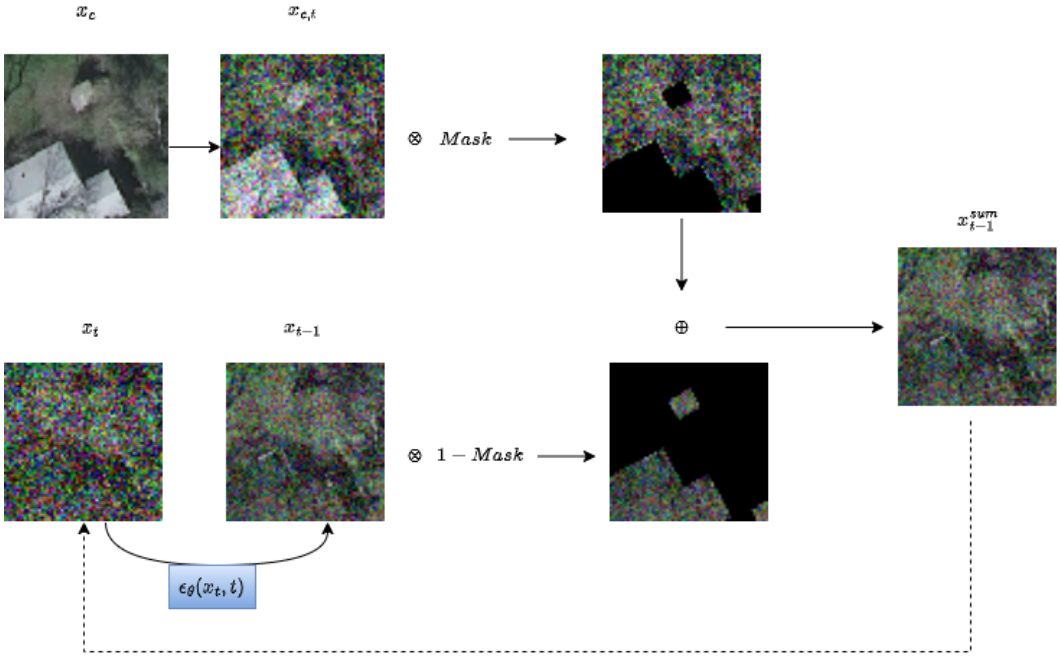


Figure 4.4. Summation conditioning approach. As we see, it is an iterative process which estimates the image

Chapter 5

Experiments

Now we will present all the experiments on the tasks we have mentioned in the introduction. In particular, we will start from a simple proof of concept of how diffusion models can learn to generate new EO data; this part will equip only the unconditional formulation, and will serve as a baseline for future refinement s and more complex applications.
In the second section, we will present a case for urban landscape replanning, in particular for the removal of present buildings into the scene. This use-case can be beneficial for further investigation about the effect of human presence in the territory. Moreover, it will provide us more insights about the behaviour of diffusion models for inpainting applications.
In the third section we show a cloud-removal application employing a novel guiding strategy. In this case, we will show how our model can respond to different types of clouds and different cloud coverage in the image.
Finally, we present these models as a valid data augmentator for some specific downstream tasks, namely change detection. As specified in the introduction, we want to provide new EO data with corresponding labels, with the objective of improving performances of other models on new synthetic data.
Finally, we present a bunch of challenging scenarios where these models expose some of their hidden weaknesses, thus opening the road to successive investigations.
All the experiments have been done on a RTX NVIDIA Quadro 4000 single GPU (49GB).

5.1 Generation of New EO Data

The first task we can propose to leverage and test our solution is the simple, unconditional generation of new EO images.

Datasets We use the Inria Aerial Labeling Image Dataset provided by [138]. The data are provided as 1024×1024 tiles with RGB images and corresponding masks, divided into 5 different regions: Tyrol, Vienna, Austin, Chicago and Kitsap County. There are in total 36 image-mask pairs for every town.
We decide to split the data into 64×64 patches with a 0.5 overlap factor. We use 306k images as a training step and 54k images as validation set.
We train the model for 20 hours, with the best results in terms of MSE showing at 5/6 hours of training.

Model architecture We adopt an l2 MSE diffusion loss, with a four layer U-Net structure with base channels 128 and kernel multiplication factors of [1,2,4,8], 1 resolution block per layer and no attention block. We provide results also for 2 resolution blocks and 1-head attention layer concatenated to the resblock.

Qualitative results In fig 5.2 and 5.1 we show some qualitative results for diffusion models with and without attention

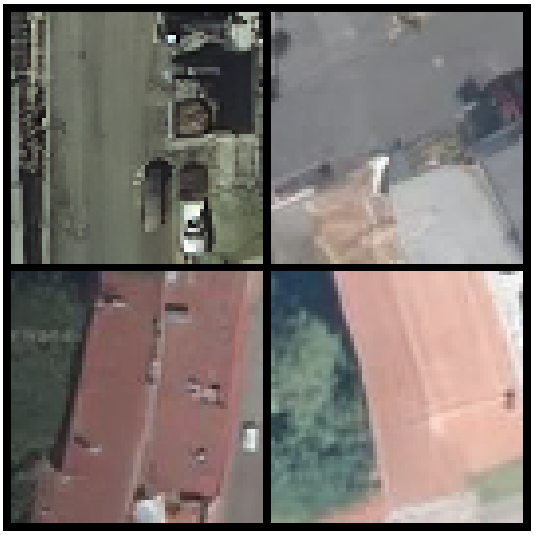


Figure 5.1. Unconditional Samples generation w/o attention

Figure 5.2. Unconditional Samples generation w/ attention

5.2 Urban Replanning

In this section we are going to present the first use-case leveraging the conditional formulation of our model. In detail, we will show how inpainting applied to urban

replanning can turn useful for specific context such as simulation of future scenarios

Datasets We use as well the Inria Aerial Labeling dataset [138] because it offers segmentation masks for the buildings we find in the dataset.
We apply the summation-based inpainting approach, therefore there is no need to retrain the network, but we can apply the conditioning during inference.

Qualitative and Quantitative evaluations We adopt two common metrics for inpainting approach: Peak-To-Signal Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM). PSNR is very useful to measure the absolute error between the inpainted image and the original one. In fact, PSNR $= 10\, log_{10}\left(\frac{MAX_I^2}{MSE}\right)$.
SSIM is more related to the properties of the image (luminance, brightness, sharpness), as we can see from the following formula: $SSIM(x,y) = \frac{(2\mu_x\mu_y+C_1)+(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)}$
where we take into account both the mean, the variances and the covariance of the two images. The key task here is to replace buildings with portions of natural environments. This approach is useful in two directions:
firtsly, we evaluate the coherence and accuracy of the model concerning the inpainting of specific objects in the image, namely the buildings;
secondly, we aim at simulate a post civilization scenario, where we forecast the evolution of nature without human presence.
In fig. 5.3 we show some examples extracted from the network area predictions In

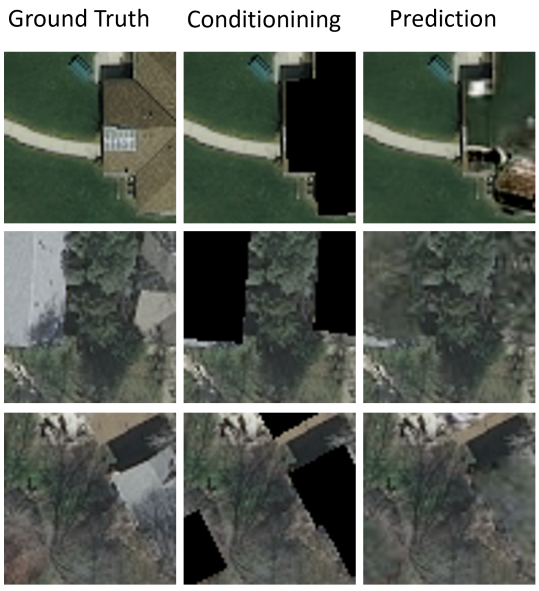


Figure 5.3. Some samples for the building removal task

table 5.1 we provide the PSNR and SSIM related to the task of building inpaint We compute the scores for Inria Aerial Dataset based on 1200 images.

Building removal metrics		
Dataset	SSIM	PSNR (DB)
InriaAerial	0.626	15.699

Table 5.1. Comparison of metrics for building removal

5.3 Cloud Removal

After validating our work on aerial images, we will test it on Sentinel-2 data. The transfer is not immediate, because in the first case we have 1-10 cm resolution, while in the second case we have to deal with lower resolution (30m) images. We test our approach on Sentinel-2 Cloud Mask Catalogue [139], a Sentinel-2 cloud mask dataset labeled on all the 13 bands of Sentinel 2.
We train and test using the RGB bands, namely band 2,3 and 4. We train on cloud-free images, and then we evaluate our approach on cloudy data samples through summation-inpainting.
Our trainining set consist in 8k 64x64 images. Our test set is composed by 167 64x64 images. We choose to focus on few categories of images (landcover, mountains-hills, urban) to avoid an excessive dispersion of training samples features to be learnt. This choice comes at the cost of few training samples, and very few test images.
We use a U-Net as estimator, with the structure decribed in chapter 4. We deploy a 4-layer network with two resolution blocks per layer, an attention block with 1 head, 128 base channels with a scaling factor of [1,2,3,4].
In table 5.2 you find the quantitative results of our approach. As you can see, we

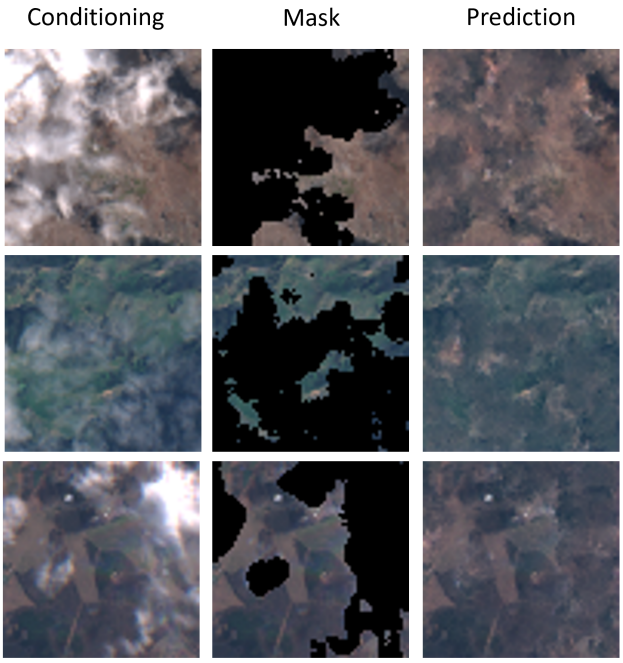


Figure 5.4. Some samples for the cloud removal task

perform slightly better on SSIM and far worse on PSNR than aerial images.
This is somehow surpising, because Sentinel-2 images should be more difficult to learn. However, this may be due to a limited size in training set, with a possible overfitting of the network. Another explanation may be the difference between the two metrics, with PSNR related to the brute pixel-wise difference between two images. This suggest us that our model has learned to reproduce the image color histogram, but still lacking pixel-precision.

Cloud removal metrics		
Dataset	SSIM	PSNR (DB)
S-2 CloudMask	0.691	24.593

Table 5.2. Comparison of metrics for cloud removal

5.4 Downstream Task Application

We provide a different application for diffusion models: as we said in the introduction, remote sensing lacks labeling for the enormous amount of data it displays. Sometimes, even in the case of label availability we may have still insufficient labels to train properly a DNN on a specific downstream task.
We choose to create a novel dataset for change detection starting from OSCD [35]. The aim of this study is to test a State-of-the-Art approach in CD on our novel dataset.
The task comprises three phases:

1. *Diffusion model training*: We train the model using a training set of 20k 64x64 images

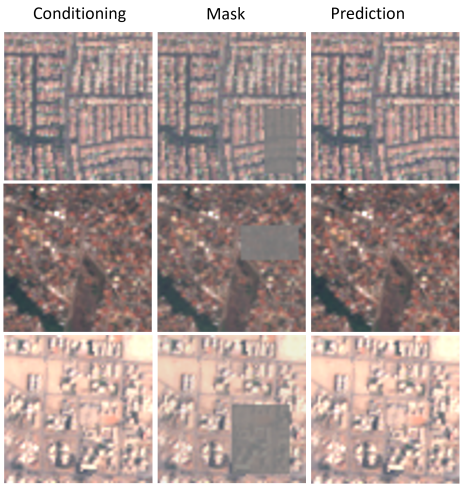
2. *Dataset generation*: we sample from our model generating new pairs image 1-image 2, where image 2 is the newly generated image from image1. We apply a random mask to image 1, then we generate a new image by inpainting the mask.

3. *Change detection approach*:

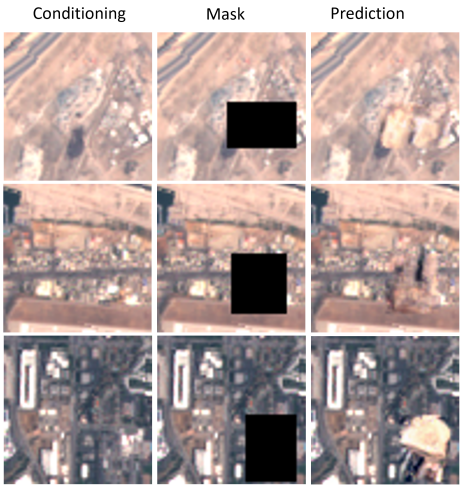
For phase 1 and 2, we report some qualitative results with a small ablation study: thin masks (some pixels in the mask are 0, some have original values) and thick masks (all pixels in the mask set as 0). We show some qualitative results in fig. 5.5a and 5.5b In table 5.3, we show some metrics computed for this dataset.

Inpainting metrics		
Dataset	SSIM	PSNR (DB)
OSCD-thin	0.997	50.375
OSCD-thick	0.831	25.363

Table 5.3. Comparison of metrics for oscd dataset generation. As we may expect, thin masks allow the model to reconstruct almost perfectly the inpainted region



(a) Oscd inpainting generation with thin masks



(b) Oscd inpaint task with thick masks

5.5 Failure Cases

Chapter 6

Conclusions

In this work, we highlighted the potential of diffusion models, a new generative approach from computer vision. We applied this formulation for a particular domain, remote sensing, with the final goal of addressing some common issues and challenges related to EO data.
We presented a comprehensive review of current diffusion models formulations, with two goals in mind:

1. Providing an ordered and structured literature review, drawing a coherent path showing the evolution of these models; we included in this path both the architectural choices behind them and the fundamental mathematical blocks behind their working principle.

2. Showing the versatility and reusability of these models by commenting some key applications in many different domains; moreover, every single different application have contributed to modifications of the original diffusion model formulation, showing both new, valuable adaptations of Diffusion Models to the specific use-case and exposing some hidden caveats.
   We presented this applications because their adaptation strategies can be source of inspiration for future researchers willing to apply these models to EO.

After that, we presented our key contribution. We have applied Diffusion Models to several EO use-cases, namely cloud removal, urban replanning and new dataset generation for downstream tasks.
We showed that Diffusion Models can be a viable option to address some typical EO challenges because they are capable to generate new EO data out of a given remote sening dataset.
But this is only a pleriminary work, and there are a lot of followups to be explored:

- *Sampling speed.* Actually, these models still have substantial limitations concerning the necessary time to generate an image; these models are still unable to generate high-quality samples for EO images with fewer timesteps. A future improvement will having a better trained model or a specific, tailored sampler for EO models to reduce the necessary number of timesteps at inference time.

- *Guidance signal.* We focused mainly on inpainting approaches, therefore the only guidance was provided by the input image itself. An additional conditioning, such as text, can improve the generation in two ways:

firstly, it adds more flexibility and control over the type of content we want to generate;
secondly, it may correct the network output in case of mode-collapse, for example by means of negative prompts.

- *Video and 3D models.* For now, we have excluded the time dimension from our formulation, which prevents us from extending our work to weather forecast scenarios. For example, an interesting application will be deploying diffusion models for cloud movement generation.
  Moreover, working on 2D data only misses an important aspect of meteorological phenomena, such as the height of a cloud and their relative depth.

- *Combination with domain adaptation and self-supervised learning.* As we underlined in the introduction, a major motivation behind these models lies in their applicability to low labeled data availability. There are still issues to adapt a pre-trained diffusion model to a different dataset, while this is critical to transfer knowledge from a rich dataset to a poorly annotated one.
  Moreover, Diffusion Models can be thought as a pseudo-label generator for self-supervised learning approaches

- *Multimodality.* There still a lot of undiscovered potential concerning fusion of different EO bands and sensors using diffusion models. We focuse mainly on the RGB bands, but Sentinel-2 has other ten bands available. Moreover, a huge step is required to adapt these models to Sentinel-1 data, which requires a careful manipulation and more context to the model, thus requiring specific conditioning strategies to incorporate the physical parameters of SAR images.

As we see, there are plenty of use-cases to be explored in the following months and years. We hope that this work will serve as an opening key to the field to foster the large-scale adaptation of these models in the context of Earth Observation.

Bibliography

[1] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[2] Geoffrey E. Hinton and Ruslan Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313:504 – 507, 2006.

[3] Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2016.

[4] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *ArXiv*, abs/1711.00937, 2017.

[5] Ali Razavi, Aäron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Neural Information Processing Systems*, 2019.

[6] Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, 2016.

[7] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[8] Irene Amerini, Leonardo Galteri, Roberto Caldelli, and A. Bimbo. Deepfake video detection through optical flow based cnn. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 1205–1207, 2019.

[9] David Guera and Edward J. Delp. Deepfake video detection using recurrent neural networks. *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, 2018.

[10] Cecilia Pasquini, Irene Amerini, and Giulia Boato. Media forensics on social media platforms: a survey. *EURASIP Journal on Information Security*, 2021:1–19, 2021.

[11] Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *ArXiv*, abs/1701.07875, 2017.

[12] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, 2018.

[13] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *ArXiv*, abs/1710.10196, 2017.

[14] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *ArXiv*, abs/2006.06676, 2020.

[15] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows, 2015.

[16] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*, 2021.

[17] Diederik P. Kingma, Tim Salimans, and Max Welling. Improved variational inference with inverse autoregressive flow. *ArXiv*, abs/1606.04934, 2016.

[18] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans. *ArXiv*, abs/2112.07804, 2021.

[19] Devis Tuia, Konrad Schindler, Begüm Demir, Gustau Camps-Valls, Xiao Xiang Zhu, Mrinalini Kochupillai, Savso Dvzeroski, Jan N. van Rijn, Holger H. Hoos, Fabio Del Frate, Mihai Datcu, Jorge-Arnulfo Quian'e-Ruiz, Volker Markl, B. L. Saux, and Rochelle Schneider. Artificial intelligence to advance earth observation: a perspective. *ArXiv*, abs/2305.08413, 2023.

[20] Zhenfeng Shao, Linjing Zhang, and Lei Wang. Stacked sparse autoencoder modeling using the synergy of airborne lidar and satellite optical and sar data to map forest above-ground biomass. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10:5569–5582, 2017.

[21] Naftaly Muriuki Wambugu, Yiping Chen, Zhenlong Xiao, Mingqiang Wei, Saifullahi Aminu Bello, José Marcato Junior, and Jonathan Li. A hybrid deep convolutional neural network for accurate land cover classification. *Int. J. Appl. Earth Obs. Geoinformation*, 103:102515, 2021.

[22] Hermann Courteille, Adams Benoît, Nicolas Méger, Abdourrahmane Mahamane Atto, and Dino Ienco. Channel-based attention for land cover classification using sentinel-2 time series. *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, pages 1077–1080, 2021.

[23] Rongjun Qin and Tao Liu. A review of landcover classification with very-high resolution remotely sensed optical images-analysis unit, model scalability and transferability. *Remote. Sens.*, 14:646, 2022.

[24] Roberto Del Prete, Maria Daniela Graziano, and Alfredo Renga. Unified framework for ship detection in multi-frequency sar images: A demonstration with cosmo-skymed, sentinel-1, and saocom data. *Remote. Sens.*, 15:1582, 2023.

[25] Bogdan Iancu, Valentin Soloviev, Luca Zelioli, and Johan Lilius. Aboships - an inshore and offshore maritime vessel detection dataset with precise annotations. *ArXiv*, abs/2102.05869, 2021.

[26] D. Spiller, Luigi Ansalone, Stefania Amici, Alessandro Piscini, and Pierre-Philippe Mathieu. Analysis and detection of wildfires by using prisma hyperspectral imagery. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2021.

[27] Yingchang Li, Mingyang Li, Chao kui Li, and Zhenzhen Liu. Forest aboveground biomass estimation using landsat 8 and sentinel-1a data with machine learning algorithms. *Scientific Reports*, 10, 2020.

[28] Jung Min Han, Yu Qian Ang, Ali Malkawi, and Holly W. Samuelson. Using recurrent neural networks for localized weather prediction with combined use of public airport data and on-site measurements. *Building and Environment*, 192:107601, 2021.

[29] Ryan Keisler. Forecasting global weather with graph neural networks. *ArXiv*, abs/2202.07575, 2022.

[30] Rémi R. Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Alexander Pritzel, Suman V. Ravuri, Timo Ewalds, Ferran Alet, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Jacklynn Stott, Oriol Vinyals, Shakir Mohamed, and Peter W. Battaglia. Graphcast: Learning skillful medium-range global weather forecasting. *ArXiv*, abs/2212.12794, 2022.

[31] Meredith L. Fowlie, Edward A. Rubin, and Reed Walker. Bringing satellite-based air quality estimates down to earth. *ERN: Environmental Economics (Topic)*, 2019.

[32] Jonathan M. Frame, Frederik Kratzert, Daniel Klotz, Martin Gauch, Guy Shelev, Oren Gilon, Logan M. Qualls, Hoshin Vijai Gupta, and Grey S. Nearing. Deep learning rainfall-runoff predictions of extreme events. *Hydrology and Earth System Sciences*, 2021.

[33] Pedram Ghamisi, Behnood Rasti, Naoto Yokoya, Qunming Wang, Bernhard Hofle, Lorenzo Bruzzone, Francesca Bovolo, Mingmin Chi, Katharina Anders, Richard Gloaguen, et al. Multisource and multitemporal data fusion in remote sensing: A comprehensive review of the state of the art. *IEEE Geoscience and Remote Sensing Magazine*, 7(1):6–39, 2019.

[34] Rodrigo Caye Daudt, B. L. Saux, and Alexandre Boulch. Fully convolutional siamese networks for change detection. *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 4063–4067, 2018.

[35] Rodrigo Caye Daudt, B. L. Saux, Alexandre Boulch, and Yann Gousseau. Urban change detection for multispectral earth observation using convolutional neural networks. *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 2115–2118, 2018.

[36] Sheng Fang, Kaiyu Li, Jinyuan Shao, and Zhe Li. Snunet-cd: A densely connected siamese network for change detection of vhr images. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2021.

[37] Jiahui Qu, Shaoxiong Hou, Wenqian Dong, Yunsong Li, and Weiying Xie. A multilevel encoder–decoder attention network for change detection in hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2022.

[38] W. G. C. Bandara and Vishal M. Patel. A transformer-based siamese network for change detection. *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, pages 207–210, 2022.

[39] Andrea Codegoni, Gabriele Lombardi, and Alessandro Ferrari. Tinycd: a (not so) deep learning model for change detection. *Neural Computing and Applications*, 35:8471 – 8486, 2022.

[40] Valerio Marsocci, Virginia Coletta, Roberta Ravanelli, Simone Scardapane, and Mattia Giovanni Crespi. Inferring 3d change detection from bitemporal optical images. *ArXiv*, abs/2205.15903, 2022.

[41] Alessandro Sebastianelli, Erika Puglisi, Maria Pia del Rosso, Jamila Mifdal, Artur Nowakowski, Pierre-Philippe Mathieu, F. Pirri, and Silvia Liberata Ullo. Plfm: Pixel-level merging of intermediate feature maps by disentangling and fusing spatial and temporal data for cloud removal. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–16, 2022.

[42] Fan Meng, Xiaomei Yang, Chenghu Zhou, and Zhi Li. A sparse dictionary learning-based adaptive patch inpainting method for thick clouds removal from high-spatial resolution remote sensing imagery. *Sensors (Basel, Switzerland)*, 17, 2017.

[43] Xinghua Li, Huanfeng Shen, Liangpei Zhang, and Huifang Li. Sparse-based reconstruction of missing information in remote sensing images from spectral/temporal complementary information. *Isprs Journal of Photogrammetry and Remote Sensing*, 106:1–15, 2015.

[44] Xinghua Li, Liyuan Wang, Qing Cheng, Penghai Wu, Wenxia Gan, and Lina Fang. Cloud removal in remote sensing images using nonnegative matrix factorization and error correction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2019.

[45] Andreas Meraner, Patrick Ebel, Xiaoxiang Zhu, and Michael Schmitt. Cloud removal in sentinel-2 imagery using a deep residual neural network and sar-optical data fusion. *Isprs Journal of Photogrammetry and Remote Sensing*, 166:333 – 346, 2020.

[46] Jianhao Gao, Qiangqiang Yuan, Jie Li, Hai Zhang, and Xin Su. Cloud removal with fusion of high resolution optical and sar images using generative adversarial networks. *Remote. Sens.*, 12:191, 2020.

[47] Teng-Yu Ji, Delin Chu, Xile Zhao, and Danfeng Hong. A unified framework of cloud detection and removal based on low-rank and group sparse regularizations for multitemporal multispectral images. *IEEE Transactions on Geoscience and Remote Sensing*, PP:1–1, 2022.

[48] Jun Yu Li, Zhaocong Wu, Qinghong Sheng, Bo Wang, Zhongwen Hu, Shaobo Zheng, Gustau Camps-Valls, and M. Molinier. A hybrid generative adversarial network for weakly-supervised cloud detection in multispectral images. *Remote Sensing of Environment*, 280, 2022.

[49] Qing Cheng, Huanfeng Shen, Liangpei Zhang, and Pingxiang Li. Inpainting for remotely sensed images with a multichannel nonlocal total variation model. *IEEE Transactions on Geoscience and Remote Sensing*, 52:175–187, 2014.

[50] Fan Meng, Xiaomei Yang, Chenghu Zhou, and Zhi Li. A sparse dictionary learning-based adaptive patch inpainting method for thick clouds removal from high-spatial resolution remote sensing imagery. *Sensors (Basel, Switzerland)*, 17, 2017.

[51] Heng Pan. Cloud removal for remote sensing imagery via spatial attention generative adversarial network. *ArXiv*, abs/2009.13015, 2020.

[52] Wen-Jie Zheng, Xile Zhao, Yu-Bang Zheng, Jie Lin, Lina Zhuang, and Ting-Zhu Huang. Spatial-spectral-temporal connective tensor network decomposition for thick cloud removal. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2023.

[53] Qiang Zhang, Qiangqiang Yuan, Jie Li, Zhiwei Li, Huanfeng Shen, and Liangpei Zhang. Thick cloud and cloud shadow removal in multitemporal imagery using progressively spatio-temporal patch group deep learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2020.

[54] Shunping Ji, Peiyu Dai, Meng Lu, and Yongjun Zhang. Simultaneous cloud detection and removal from bitemporal remote sensing images using cascade convolutional neural networks. *IEEE Trans. Geosci. Remote. Sens.*, 59:732–748, 2021.

[55] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[56] Vishnu Sarukkai, Anirudh Jain, Burak Uzkent, and Stefano Ermon. Cloud removal from satellite images using spatiotemporal generator networks. 2020.

[57] Cengis Hasan, Ross Horne, Sjouke Mauw, and Andrzej Mizera. Cloud removal from satellite imagery using multispectral edge-filtered conditional generative adversarial networks. *International Journal of Remote Sensing*, 43:1881 – 1893, 2022.

[58] Praveer Singh and Nikos Komodakis. Cloud-gan: Cloud removal for sentinel-2 imagery using a cyclic consistent generative adversarial networks. *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 1772–1775, 2018.

[59] Patrick Ebel, Andreas Meraner, Michael Schmitt, and Xiaoxiang Zhu. Multi-sensor data fusion for cloud removal in global and all-season sentinel-2 imagery. *ArXiv*, abs/2009.07683, 2020.

[60] Patrick Ebel, Yajin Xu, Michael Schmitt, and Xiaoxiang Zhu. Sen12ms-cr-ts: A remote sensing data set for multi-modal multi-temporal cloud removal. *IEEE Transactions on Geoscience and Remote Sensing*, PP:1–1, 2022.

[61] Wei He and Naoto Yokoya. Multi-temporal sentinel-1 and -2 data fusion for optical image simulation. *ISPRS Int. J. Geo Inf.*, 7:389, 2018.

[62] Wei He and Naoto Yokoya. Multi-temporal sentinel-1 and -2 data fusion for optical image simulation. *ISPRS Int. J. Geo Inf.*, 7:389, 2018.

[63] Faramarz Naderi Darbaghshahi, Mohammad Reza Mohammadi, and Mohsen Soryani. Cloud removal in remote sensing images using generative adversarial networks and sar-to-optical image translation. *IEEE Transactions on Geoscience and Remote Sensing*, PP:1–1, 2020.

[64] Corinne Stucker, Vivien Sainte Fare Garnot, and Konrad Schindler. U-tilise: A sequence-to-sequence model for cloud removal in optical satellite time series. *ArXiv*, abs/2305.13277, 2023.

[65] Vivien Sainte Fare Garnot and Loic Landrieu. Panoptic segmentation of satellite image time series with convolutional temporal attention networks. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4852–4861, 2021.

[66] Patrick Ebel, Vivien Sainte Fare Garnot, Michael Schmitt, Jan Dirk Wegner, and Xiao Xiang Zhu. Uncrtaints: Uncertainty quantification for cloud removal in optical satellite time series. *ArXiv*, abs/2304.05464, 2023.

[67] Jakob Gawlikowski, Patrick Ebel, Michael Schmitt, and Xiao Xiang Zhu. Explaining the effects of clouds on remote sensing scene classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:9976–9986, 2022.

[68] Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128:336–359, 2016.

[69] Lilian Weng. What are diffusion models? *lilianweng.github.io*, Jul 2021.

[70] Yang Song, Jascha Narain Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *ArXiv*, abs/2011.13456, 2020.

[71] Jascha Narain Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *ArXiv*, abs/1503.03585, 2015.

[72] Brian. D. O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12:313–326, 1982.

[73] Simo Särkkä and A. Solin. Applied stochastic differential equations. 2019.

[74] Bernt Øksendal. Stochastic differential equations : an introduction with applications. *Journal of the American Statistical Association*, 82:948, 1987.

[75] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *ArXiv*, abs/1907.05600, 2019.

[76] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *ArXiv*, abs/2006.09011, 2020.

[77] Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *ArXiv*, abs/2107.00630, 2021.

[78] Jonathan Ho, Ajay Jain, and P. Abbeel. Denoising diffusion probabilistic models. *ArXiv*, abs/2006.11239, 2020.

[79] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *ArXiv*, abs/2206.00364, 2022.

[80] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ArXiv*, abs/2010.02502, 2020.

[81] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *ArXiv*, abs/2202.00512, 2022.

[82] Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *ArXiv*, abs/2302.04867, 2023.

[83] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *ArXiv*, abs/2211.01095, 2022.

[84] Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie Li, Hamideh Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise. *ArXiv*, abs/2208.09392, 2022.

[85] Andrew Campbell, Joe Benton, Valentin De Bortoli, Tom Rainforth, George Deligiannidis, and A. Doucet. A continuous time framework for discrete denoising models. *ArXiv*, abs/2205.14987, 2022.

[86] Valentin De Bortoli, James Thornton, Jeremy Heng, and A. Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. In *Neural Information Processing Systems*, 2021.

[87] Valentin De Bortoli. Convergence of denoising diffusion models under the manifold hypothesis. *Trans. Mach. Learn. Res.*, 2022, 2022.

[88] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *ArXiv*, abs/2105.05233, 2021.

[89] Jonathan Ho. Classifier-free diffusion guidance. *ArXiv*, abs/2207.12598, 2022.

[90] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. *ArXiv*, abs/2302.07121, 2023.

[91] William S. Peebles and Saining Xie. Scalable diffusion models with transformers. *ArXiv*, abs/2212.09748, 2022.

[92] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *ArXiv*, abs/1505.04597, 2015.

[93] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. In *Neural Information Processing Systems*, 2021.

[94] Bowen Jing, Gabriele Corso, Renato Berlinghieri, and Tommi Jaakkola. Subspace diffusion generative models. In *European Conference on Computer Vision*, 2022.

[95] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2021.

[96] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. *ArXiv*, abs/2204.03458, 2022.

[97] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey A. Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models. *ArXiv*, abs/2210.02303, 2022.

[98] Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23:47:1–47:33, 2021.

[99] Eyal Molad, Eliahu Horwitz, Dani Valevski, Alex Rav Acha, Y. Matias, Yael Pritch, Yaniv Leviathan, and Yedid Hoshen. Dreamix: Video diffusion models are general video editors. *ArXiv*, abs/2302.01329, 2023.

[100] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *ArXiv*, abs/2211.11018, 2022.

[101] Johanna Suvi Karras, Aleksander Holynski, Ting-Chun Wang, and Ira Kemelmacher-Shlizerman. Dreampose: Fashion image-to-video synthesis via stable diffusion. *ArXiv*, abs/2304.06025, 2023.

[102] Uriel Singer, Adam Polyak, Thomas Hayes, Xiaoyue Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. *ArXiv*, abs/2209.14792, 2022.

[103] Shengqu Cai, Eric Chan, Songyou Peng, Mohamad Shahbazi, Anton Obukhov, Luc Van Gool, and Gordon Wetzstein. Diffdreamer: Consistent single-view perpetual view generation with conditional diffusion models. *ArXiv*, abs/2211.12131, 2022.

[104] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *ArXiv*, abs/2209.14988, 2022.

[105] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. *ArXiv*, abs/2211.10440, 2022.

[106] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts, 2022.

[107] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2836–2844, 2021.

[108] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, 2021.

[109] Uriel Singer, Shelly Sheynin, Adam Polyak, Oron Ashual, Iurii Makarov, Filippos Kokkinos, Naman Goyal, Andrea Vedaldi, Devi Parikh, Justin Johnson, and Yaniv Taigman. Text-to-4d dynamic scene generation. *ArXiv*, abs/2301.11280, 2023.

[110] Rishabh Dabral, Muhammad Hamza Mughal, Vladislav Golyanik, and Christian Theobalt. Mofusion: A framework for denoising-diffusion-based motion synthesis. *ArXiv*, abs/2212.04495, 2022.

[111] Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, Jingyi Yu, and Gang Yu. Executing your commands via motion diffusion in latent space. *ArXiv*, abs/2212.04048, 2022.

[112] Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. Remodiffuse: Retrieval-augmented motion diffusion model. *ArXiv*, abs/2304.01116, 2023.

[113] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H. Bermano. Human motion diffusion model. *ArXiv*, abs/2209.14916, 2022.

[114] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *ArXiv*, abs/2003.08934, 2020.

[115] Titas Anciukevicius, Zexiang Xu, Matthew Fisher, Paul Henderson, Hakan Bilen, Niloy Jyoti Mitra, and Paul Guerrero. Renderdiffusion: Image diffusion for 3d reconstruction, inpainting and generation. *ArXiv*, abs/2211.09869, 2022.

[116] Cristian Sbrolli, Paolo Cudrano, Matteo Frosi, and Matteo Matteucci. Ic3d: Image-conditioned 3d diffusion for shape generation. *ArXiv*, abs/2211.10865, 2022.

[117] Jamie M. Wynn and Daniyar Turmukhambetov. Diffusionerf: Regularizing neural radiance fields with denoising diffusion models. *ArXiv*, abs/2302.12231, 2023.

[118] Jiatao Gu, Alex Trevithick, Kai-En Lin, Joshua M. Susskind, Christian Theobalt, Lingjie Liu, and Ravi Ramamoorthi. Nerfdiff: Single-image view synthesis with nerf-guided distillation from 3d-aware diffusion. *ArXiv*, abs/2302.10109, 2023.

[119] Norman Muller, Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Peter Kontschieder, and Matthias Nießner. Diffrf: Rendering-guided 3d radiance field diffusion. *ArXiv*, abs/2212.01206, 2022.

[120] Zhen Liu, Yao Feng, Michael J. Black, Derek Nowrouzezahrai, Liam Paull, and Wei yu Liu. Meshdiffusion: Score-based generative 3d mesh modeling. *ArXiv*, abs/2303.08133, 2023.

[121] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. *ArXiv*, abs/2210.06978, 2022.

[122] Willi Menapace, Aliaksandr Siarohin, Stéphane Lathuilière, Panos Achlioptas, Vladislav Golyanik, Elisa Ricci, and S. Tulyakov. Plotting behind the scenes: Towards learnable game engines. *ArXiv*, abs/2303.13472, 2023.

[123] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur D. Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34:18–42, 2016.

[124] Clément Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. *ArXiv*, abs/2209.14734, 2022.

[125] Zhiqing Sun and Yiming Yang. Difusco: Graph-based diffusion solvers for combinatorial optimization. *ArXiv*, abs/2302.08224, 2023.

[126] Bowen Jing, Gabriele Corso, Jeffrey Chang, Regina Barzilay, and T. Jaakkola. Torsional diffusion for molecular conformer generation. *ArXiv*, abs/2206.01729, 2022.

[127] Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and T. Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking. *ArXiv*, abs/2210.01776, 2022.

[128] Peiye Zhuang, Samira Abnar, Jiatao Gu, Alex Schwing, Joshua M. Susskind, and Miguel 'Angel Bautista. Diffusion probabilistic fields. *ArXiv*, abs/2303.00165, 2023.

[129] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *ArXiv*, abs/2009.09761, 2020.

[130] Qingqing Huang, Daniel S. Park, Tao Wang, Timo I. Denk, Andy Ly, Nanxin Chen, Zhengdong Zhang, Zhishuai Zhang, Jiahui Yu, C. Frank, Jesse Engel, Quoc V. Le, William Chan, and Weixiang Han. Noise2music: Text-conditioned music generation with diffusion models. *ArXiv*, abs/2302.03917, 2023.

[131] Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matthew Sharifi, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. Audiolm: a language modeling approach to audio generation. *ArXiv*, abs/2209.03143, 2022.

[132] Giorgio Mariani, Irene Tallini, Emilian Postolache, Michele Mancusi, Luca Di Cosmo, and Emanuele Rodolà. Multi-source diffusion models for simultaneous music generation and separation. *ArXiv*, abs/2302.02257, 2023.

[133] Ludan Ruan, Y. Ma, Huan Yang, Huiguo He, Bei Liu, Jianlong Fu, Nicholas Jing Yuan, Qin Jin, and Baining Guo. Mm-diffusion: Learning multi-modal diffusion models for joint audio and video generation. *ArXiv*, abs/2212.09478, 2022.

[134] Dan Bigioi, Shubhajit Basak, Hugh Jordan, Rachel Mcdonnell, and Peter Corcoran. Speech driven video editing via an audio-conditioned diffusion model. *ArXiv*, abs/2301.04474, 2023.

[135] Yichong Leng, Zehua Chen, Junliang Guo, Haohe Liu, Jiawei Chen, Xu Tan, Danilo P. Mandic, Lei He, Xiang-Yang Li, Tao Qin, Sheng Zhao, and Tie-Yan Liu. Binauralgrad: A two-stage conditional diffusion probabilistic model for binaural audio synthesis. *ArXiv*, abs/2205.14807, 2022.

[136] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *ArXiv*, abs/2105.05233, 2021.

[137] Andreas Lugmayr, Martin Danelljan, Andrés Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11451–11461, 2022.

[138] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3226–3229, 2017.

[139] Alistair Francis, John Mrziglod, Panagiotis Sidiropoulos, and Jan-Peter Muller. Sentinel-2 cloud mask catalogue, November 2020.