A&A, 686, A285 (2024)
https://doi.org/10.1051/0004-6361/202347860

Astronomy & Astrophysics

Solar synthetic imaging: Introducing denoising diffusion probabilistic models on SDO/AIA data

F. P. Ramunno[1,2], S. Hackstein[1], V. Kinakh[2], M. Drozdova[2], G. Quétant[2], A. Csillaghy[1], and S. Voloshynovskiy[2]

[1] Institute for Data Science, University of Applied Sciences North Western Switzerland (FHNW), 5210 Windisch, Switzerland
e-mail: francesco.ramunno@fhnw.ch
[2] Department of Computer Science, University of Geneva, 1211 Geneva, Switzerland

Received 1 September 2023 / Accepted 9 March 2024

ABSTRACT

For the luck of humanity, there are way less big solar flares than small ones. Even if these are good news, this makes it challenging to train machine learning algorithms able to model solar activity. As a result, solar monitoring applications, including flare forecasting, suffer from this lack of input data. To overcome this issue, generative deep learning models can be utilised to produce synthetic images representing solar activity and thus compensating the rarity of big events. This study aims to develop a method that can generate synthetic images of the Sun with the ability to include flare of a specific intensity. To achieve our goals, we introduce a Denoising Diffusion Probabilistic Model (DDPM). We train it with a carefully crafted dataset from the Atmospheric Image Assembly (AIA) instrument on the SDO spacecraft, specifically the 171 Å band, which captures images of coronal loops, filaments, flares, and active regions. GOES X-ray measurements are employed to classify each image based on the solar flare scale (A, B, C, M, X), after selecting the flaring images from AIA using the Heliophysics Event Knowledgebase, which allows for temporal localisation of the flaring events. The generative model performance is evaluated using cluster metrics, Fréchet Inception Distance (FID), and the F1-score. We demonstrate state-of-the-art results in generating solar images and conduct two experiments that use the synthetic images. The first experiment trains a supervised classifier to identify those events. The second experiment trains a basic solar flare predictor. The experiments demonstrate the effectiveness of additional synthetic samples to addressing the problem of imbalanced datasets. We believe this is only the beginning of DDPM use with solar data. It remains to gain a better understanding of the generation capabilities of the denoising diffusion probabilistic models in the contest of solar flare predictions and apply them to other deep learning and physical tasks, such as AIA to HMI () image translation.

Key words. methods: data analysis – Sun: activity – Sun: flares

1. Introduction

Solar flares pose a threat to Earth and its inhabitants due to their ability to induce geomagnetic storms that can disrupt modern technological infrastructure. Their effects can have significant consequences for various technologies, such as the communication systems, causing radio communication disruptions, especially at high frequencies. This can impact on airline communications and those of emergency services and others (Knipp et al. 2016; Redmon et al. 2018; Xu et al. 2023). Solar flares are also dangerous for astronaut safety (Smith & Scalo 2007; Fargion et al. 2019) increasing their risk of radiation-related health issues and also for satellite operations (Gopalswamy et al. 2023) leading to temporary loss of service due to the increase in radiation.

This implies a need to build forecasting and nowcasting algorithms for the prediction of their arrival and mitigation or nullification of their effects (Cicogna et al. 2021; Guastavino et al. 2023; Huwyler & Melchior 2022; Tlatov & Pevtsov 2023). However, as we know, algorithms are only as good as the data they rely on.

A major problem in the prediction of solar flares begins with the fact that the intensity of solar flares is inversely proportional to their occurrence rate; indeed the most dangerous are the rarest events (Aschwanden & Freeland 2012). This results in unbalanced datasets (Wan et al. 2021), which create significant challenges in effectively training an algorithm to predict these events. The lack of generalisation and strong bias towards more frequent flare classes can be attributed to the difficulty in obtaining datasets that equally represent the various classes. For a model to successfully grasp the required information and make accurate predictions, it is essential to have datasets that represent each class equally in order to avoid biases and improve generalisation. Furthermore, understanding flares is also of interest to studies of also interest particle acceleration, plasma ejection and their morphology in different wavelengths (Battaglia et al. 2023; Collier et al. 2023). Thus, being able to study the highest energy flares with a large amount of data and with the ability to control all the characteristics of images of the Sun can also be useful in order to better understand the triggers that lead to these high energetic events and also their evolution.

Recently, there has been an increase in the popularity of generative models (Rombach et al. 2021; Ramesh et al. 2022). Consequently, it is interesting to explore the feasibility of training a model capable of recognising and generating the different patterns that define solar activities. Such a task holds the potential to change the utilisation of synthetic data, extending beyond just class representation, and facilitating the discovery of novel physics.

This work focuses on the development of a method that can generate synthetic images of the Sun, whilst allowing the user to control the presence of a flare of a given intensity. Our aim is to investigate the capacity of the model to distinguish the solar features that can potentially trigger such events and to be able to generate them.

There have been various attempts to use deep learning generative models (Liu & Carande 2022; Deng et al. 2021; Dash et al. 2022), mainly for image-to-image translation purposes (Salvatelli et al. 2022; e.g. AIA (Atmospheric Image Assembly) to HMI (Helioseismic and Magnetic Imager)). In recent years, Generative Adversarial Network (GAN) has been the state-of-the-art model for image generation and variations of this task (e.g. image-to-image translation and image inpainting Chen et al. 2022). Unfortunately, GANs present some limitations. The most important for our work is the fact that GANs suffer significantly from mode collapse. Thus, if some classes are under-represented, it is more likely that the model is going to ignore them with a preference for the most populated classes. This is why we turn our attention here to the denoising diffusion probabilistic models (DDPMs; Ho et al. 2020). Dhariwal & Nichol (2021) analysed how diffusion models can overcome the GANs limitations. The latent space learned by a diffusion model has been shown to be useful in discriminative tasks such as classification and anomaly detection (Zimmermann et al. 2021; Wolleb et al. 2022). As a result, the image quality results obtained with diffusion models are better than those with GANs as shown in Rombach et al. (2021). Furthermore, most importantly, the diffusion models are better in capturing the ground-truth distribution of the data analysed by metrics such as the FID (Fréchet inception distance; Heusel et al. 2017), which helps in cases where there are under-represented classes.

In the present work, we investigate the capabilities of the DDPMs, which have already proven to be valuable in other, diverse application domains such as computer science, medicine and astrophysics (Um et al. 2023; Huy & Quan 2023; Karchev et al. 2022). This method allows us to generate synthetic images of the Sun given a specific label from the GOES classification system . The labels are used to guide the process during the sampling towards the generation of a specific image of the Sun with the correct amount of activity.

To the best of our knowledge, we are the first to introduce the concept of DDPMs in the field of heliophysics and the first to guide the sampling process being able to fill the unbalanced high energy solar flare classes (e.g., M- and X-flare class).

We use images obtained by the Solar Dynamics Observatory (SDO) telescope in the training procedure. As a future work, with the results of this project, we aim to demonstrate the use of the synthetic images of a particular flare class to train machine learning algorithms for image classification and flare forecasting/nowcasting and to investigate these phenomena more extensively based on more available data.

This paper is organised as follows. In Sect. 2 we introduce the datasets used. In Sect. 3 we explain the DDPM together with the classifier free guidance technique. In Sect. 4 we analyse our setup and our experiments; we discuss then their results them in Sect. 6. We present two different uses of the model in Sect. 8 and finally conclude in Sect. 9.

2. Dataset

In this work, we use three datasets: (1) the version 2 of the SDO Machine Learning Dataset (SDOMLv2), which is an update of version 1 by Galvez et al. (2019), available at a dedicated Github repository[1], and provides full Sun images; (2) the GOES X-ray sensor data, which we use to retrieve the X-ray emission; and (3) the Heliophysics Events Knowledgebase (Hurlburt et al. 2010, HEK), which we use as event recording notifier.

2.1. SDO machine learning dataset

The origin of the data used is the AIA (Lemen et al. 2012), an instrument on board of the SDO satellite. AIA records full-disc images of the solar photosphere, chromosphere, and corona in two ultraviolet (UV) channels and seven extreme ultraviolet (EUV) channels. However, the AIA data cannot be used directly for ML; first they need to be preprocessed to be spatially coregistered, to have equal angular resolutions, and to be corrected for instrumental effects. Therefore, a subset called SDOML (Galvez et al. 2019) has been created so that it can be directly used for machine learning studies. In this study, we are using SDOMLv2, which is updated to account for a change in calibration after 2019, uses of the new zarr format, and adds the data up to the present day.

In this study, we are working with 64×64 images, because of the heavy computation of the model as explained in Sect. 3. We are conscious that the image size will need to be increased for an operational study. As we show in Sect. 6, the 64×64 images are still able to model the solar activity; however, in a future study, we would like to explore also the impact of image size on the applicability of the synthetic images. We are using the AIA 171 Å channel. This band is chosen because a broad range of solar activity is visible there , with many features, and therefore it is interesting to test whether or not if the generative model is able to reproduce this activity due to the complicated nature of this channel. In addition, the 171 Å channel is also used to compare the results with the work by Giger (2022), who also used this channel for an anomaly-detection task based on a generative model. This allows us to determine the ability of the DDPMs to generate images of the Sun and to examine their quality.

2.2. GOES X-Ray sensor

Since 1986, a series of GOES spacecraft have been taking measurements of soft X-rays in two energy bands (X-Ray sensor A (XRSA) 0.5–4 Å and XRSB 1–8 Å). The XRSB channel is used to mointor the solar flares and to determine their magnitude. We downloaded the data from 2011 to 2019 with the Python library SunPy (The SunPy Community 2020). Based on the intensity of the X-ray emission in W/m^2, it is possible to define a logarithmic scale with which to classify solar flares (NOAA 2023). This scale is composed of five main classes: A, B, C, M, and X with different subclasses based on the strength of the flare. The intensity of the X-ray emission of an A-class flare is less than 10^{-7} W/m^2, that of a B-class flare is 10^{-7}–10^{-6} W/m^2, and that of an X-class flare is of 10^{-4} W/m^2 or more.

2.3. Heliophysics Events Knowledgebase (HEK)

The HEK (Hurlburt et al. 2010) is a platform developed to better organise and make more efficient use of the data in the heliophysics field. We used the HEK to obtain all the peak times of flaring events from 2011 to 2019.

[1] https://sdoml.github.io

2.4. Data selection

As described in Sect. 1, the purpose of this study is to investigate whether we can train a model to generate high-energy flares filling the lower populated classes with synthetic solar images and thus creating a balanced data set. Therefore, we do not intend to build a generative model capable of arbitrarily generating an image of the sun characterised by random activity; rather, we aim to generate images of the Sun with a particular class of flare. In order to accomplish this, we must provide our model with data that depict flaring events and are labelled so that the intensity of the flaring event can be determined.

To generate this data set, we proceeded in three steps. First, we setup the access to the HEK flaring events tabular dataset. Although we use all SDOMLv2 data from 2011 to 2019, we need to take into consideration the time gap of 6 min between each image (as opposed to 12 s in the original SDO data). Using the SunPy (The SunPy Community 2020) library to connect to the HEK, we first retrieved all flaring events from 2011 to 2019, which total 107 709 distinct flaring events.

The second step was to associate the HEK flare events with their GOES class. Of the 107 709 HEK flaring events, 15 696 are already associated with a flare class. For the remaining events, we first used SunPy to access the X-ray emission values recorded by GOES/XRSB along with the time of observation. We then associated each flare peak time of the HEK event with the GOES X-ray emission closest in time, so that the values from GOES are always recorded within a time range of less than 6 s after the flare. We used the time after the peak because the decay of the X-ray emission after the flare is less steep than the increase prior to the flare, resulting in more accurate information. Up to this point, we had a dataset of 51 374 events characterised by flare class based on the X-ray emission value, the peak time of the flare, and the observation time of the GOES emission.

The third step was to correlate the HEK/GOES associated information with the SDOMLv2 data. We can obtain the observation time of each AIA image, allowing us to link the two datasets. Indeed, for every flaring peak time, we associated the closest image in time such that the image always follows the flare within a 7 min tolerance. The tolerance of 7 min is based on the fact that the time delay from SDOMLv2 is 6 min, and we want to maximise the number of images while ensuring the most accurate labelling possible due to data constraints.

As a result, we finally obtained a new set of 20 420 AIA images that are precisely labelled with their GOES flare class. We do not include images without flaring events, because we want to simulate and let the model understand the configuration of these high-energy events at all the levels. In addition, we use full-disc images because we want to test whether or not the model is able to find location of the activity thanks to the attention and convolution layers present in the diffusion model backbone as described in Sect. 4.

2.5. Limitation of the new dataset

The most significant limitation of our new dataset is the time delay. This delay is caused by the 6 min time cadence in SDOMLv2. All our results account for this, and a future enhancement could be implemented to mitigate this effect. More details and figures are analysed in Appendix A.

3. Background

In this section, we briefly introduce the DDPMs presented in (Ho et al. 2020) and their extensions to conditional generation with the classifier-free guidance (Ho & Salimans 2022, CFG) on which our study is based.

3.1. Denoising diffusion probabilistic models

A diffusion model (Ho et al. 2020; Sohl-Dickstein et al. 2015) is a type of generative model defined by a forward process – also called diffusion process – and a reverse process.

The forward process, described in Eq. (1), gradually pushes the samples off the data manifold, turning them into noise. This process is a fixed Markov chain, which gradually adds Gaussian noise and is parameterised by a variance schedule $\beta_1, ..., \beta_T$ with $\beta_t \in (0, 1)\ \forall t$ and $\beta_1 < \beta_2 < ... < \beta_T$, where T is the total number of steps of the Markov chain:

$$q(x_{1:T}|x_0) := \prod_{t=1}^{T} q(x_t|x_{t-1}),$$
$$q(x_t|x_{t-1}) := \mathcal{N}(x_t;\ \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}), \tag{1}$$

where $\mathcal{N}(a; c)$ denotes a Gaussian distribution with mean 'a' and covariance matrix 'c'. In the limit of T approaching infinity $q(x_T|x_0) \sim \mathcal{N}(0, I)$, where I is the identity matrix. The objective of the model is to determine

$$p_\theta(x_0) := \int p_\theta(x_{0:T})\mathrm{d}x_{1:T}, \tag{2}$$

$p_\theta(x_{0:T})$ is the reverse process defined as:

$$p_\theta(x_{0:T}) := p(x_T) \prod_{t=1}^{T} p(x_{t-1}|x_t),$$
$$p(x_{t-1}|x_t) := \mathcal{N}(x_t; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \tag{3}$$

where $p(x_T) = \mathcal{N}(x_T, 0, \mathbf{I})$, $\mu_\theta(x_t, t)$ is the predicted mean and $\Sigma_\theta(x_t, t)$ is the predicted covariance matrix. The reverse process is trained to produce the trajectory back from noise to the data manifold.

In order to calculate Eq. (2), we have to marginalise over all the possible trajectories $\mathrm{d}x_{1:T}$, which is intractable in this form; however we can optimise a variational lower bound on the negative log-likelihood:

$$\mathbb{E}[-\log(p_\theta(x_0))] \le \mathbb{E}\left[-\log\frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right]$$
$$= \mathbb{E}_q\left[-\log(p(x_T)) - \sum_{t\ge 1}\log\frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}\right] := L,$$

which can be rewritten as:

$$\mathbb{E}_q\Bigg[D_{\mathrm{KL}}(q(x_T|x_0)\|p(x_T))$$
$$+ \sum_{t>1} D_{\mathrm{KL}}(q(x_{t-1}|x_t, x_0\|p_\theta(x_{t-1}|x_t)) - \log(p_\theta(x_0|x_1))\Bigg],$$

where $D_{\mathrm{KL}}(\cdot\|\cdot)$ denotes the Kullback-Leibler (KL) divergence. Working with Gaussians, the KL divergences in the previous equation can be calculated in closed form; as suggested in Ho et al. (2020), we use the reverse process pasteurisation:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right), \tag{4}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$.
In conclusion, as in Ho et al. (2020), we treat the covariance matrix of Eq. (3) as a fixed hyper-parameter and we work on the mean, resulting in a simplified loss function:

$$L(\theta) = \mathbb{E}_{x_0,\epsilon\sim\mathcal{N}(0,I)}\left[\|\epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t) - \epsilon\|_2^2\right] \quad (5)$$

where $t \sim \mathcal{U}(1, ..., T)$, $\mathcal{U}$ is the uniform distribution, ϵ is the noise added to the image in the forward process and ϵ_θ is the noise predicted by the model.

3.2. Classifier-free guidance

Classifier-free guidance was introduced by Ho & Salimans (2022) to ease the process of conditioning the generation models. It has the same effect as classifier guidance (Dhariwal & Nichol 2021), but requires no training of a classifier.
The target is to change the ϵ_θ into

$$\hat{\epsilon}_\theta(x_t, c) = (1 + w)\epsilon_\theta(x_t, c) - w\epsilon_\theta(x_t), \quad (6)$$

where w is the CFG scale and $\hat{\epsilon}_\theta$ is the conditioned noise predicted as a linear interpolation between the guided prediction and the unguided prediction; the guidance is denoted by 'c'. As a result, the model is jointly trained with and without conditions based on a probability set as a hyperparameter, as described in Algorithm 2 of Ho & Salimans (2022).

4. Methodology and experiments

The main aim of these experiments is to find the most adapted setup and labelling system to generate full-disc solar images that can be used for further scientific studies and downstream applications. The generated synthetic solar images should feature a flare that corresponds to the class specified by a label.

The backbone of the architecture is a DDPM (Ho et al. 2020). The DDPM consists of a U-Net (Ronneberger et al. 2015), which is an encoder-decoder network with skip connections where the input and the output shapes are the same. More details on the architecture are given in Appendix B. We train for a total of 500 epochs using the AdamW (Loshchilov & Hutter 2017) optimiser, the mean square error (MSE) loss function, a learning rate of 3×10^{-4}, a batch size of 12 and one NVIDIA TITAN X graphics processing unit (GPU).

To better visualise the performance of the training process, we use the peak signal-to-noise ratio (PSNR), as an evaluation metric.

The model is implemented with the PyTorch framework (Paszke et al. 2019). The image resolution is 64×64 pixel for computational constraints, although we trained a DDPM with an image size of 128×128 pixel to examine the capabilities of the model if we increase the detail. More information on this experiment is given in Appendix C.

We trained three models, all of them conditioned to control the amount of generated solar activity present in the image. With this strategy, the specific flare information is encoded in the model, because we train it with this specific supervision. The distinction between the three models is based on the way we condition (or guide) them:

- Discrete labels: GOES classes A, B, C, M, and X,
- Continuous labels: GOES X-ray emission value,
- Latent space features of an encoder.

For the first model, we train the DDPM with the CFG technique as explained in Sect. 3.2. This is straightforward because every image of the dataset is labelled by one of the GOES classes: A, B, C, M and X (Sect. 2.4).

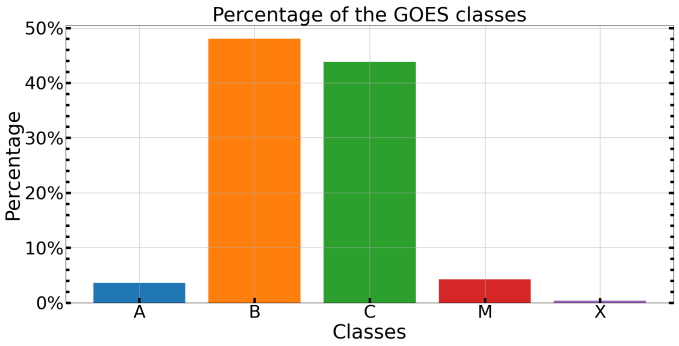

Fig. 1. Histogram distribution of the labelled dataset with the discrete GOES labels: A, B, C, M and X.

The data distribution in Fig. 1 is different from the natural distribution of the occurrence of solar flares, where flare distribution functions are successfully modelled using tapered power-law or gamma-function distributions (Sakurai 2023). This is not due to how we select data in SDOMLv2, but rather, on the one hand, to the instrumental effect of the GOES spacecraft and, on the other hand, to the threshold of the HEK catalogue, because A-class flare emission is similar to the background emission and so they are not registered as flaring events. To guide the generation and encode the label information, we use an embedding layer, where the size of the dictionary of embeddings is equal to the number of discrete classes and the size of each embedding vector is equal to the size of the time-step embeddings.

For the second model, we guide the diffusion directly with the X-ray continuous values obtained from the GOES spacecraft, as explained in Sect. 2.4. This strategy is designed to teach the model the differences between flares of different classes, avoiding the somewhat arbitrary repartition of flares into classes (e.g. a large B flare is more similar to a small C flare than to a small B flare). This way, we are able to better parameterise the class boundaries.

To guide the generation, we take the X-ray value, encode it with a sequence of two linear layers up to when the dimensions of the value are the same as those of the time-step embedding and then we sum them up.

For the third model, we guide the diffusion with the discrete labels as in our first model, but wee also add the feature embeddings of a context-encoder variational autoencoder (ceVAE) already pretrained on SDO data (Giger 2022; Zimmerer et al. 2018). The ceVAE architecture combines a Context Encoder (CE) and a Variational Autoencoder (VAE). A CE is a type of deep learning model that is trained to reconstruct an input image after randomly masking local patches of it. On the other hand, a VAE is a type of generative model that simultaneously learns the representation of the input data and its probabilistic distribution. The VAE assumes that the distribution of the latent space is Gaussian. As analysed in Zimmerer et al. (2018), the CE and the VAE are trained together, and share the same weights for the encoder-decoder architecture.

A sketch of our network is presented in Fig. 2. This procedure is designed to prepare the DDPM, giving it compressed information on the Sun with a specific amount of activity.

To guide the generation, we rely on a ceVAE that has been pretrained on SDO data, and train a DDPM to generate new ceVAE-like embeddings containing compressed information on

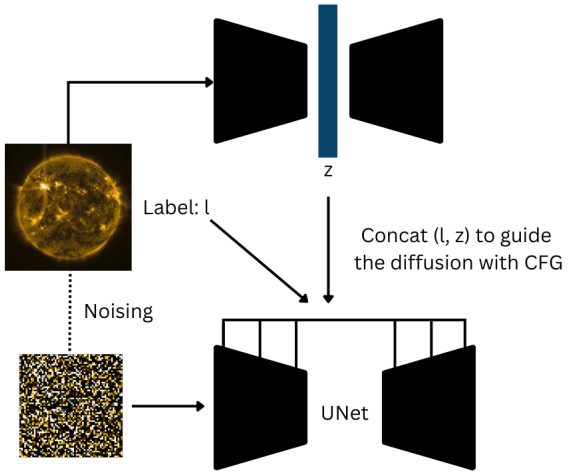

Fig. 2. Sketch of the network trained with the discrete labels and the ceVAE embeddings to guide the diffusion. In the concatenation process l represents the discrete label and z the features of the ceVAE latent space.

the Sun with a specific level of activity, as described in Dall-E 2 (Ramesh et al. 2022).

In conclusion, we use the ceVAE architecture as a baseline for our three models. The training details and performance of this architecture can be found in Giger (2022).

5. Metrics

The metrics involved in determining the quality of our models are the cluster metrics (Hackstein et al. 2023), which evaluate whether or not the generative model can produce data with the same distribution as true data without mode collapse, the Fréchet inception distance (FID) (Heusel et al. 2017), which determines the quality of generated images as well as the quality of the generated distribution; and the F1 score (Dempster et al. 1977).

The F1 score is calculated based on the precision and recall of a classification model. The precision is the number of true-positive predictions divided by the total number of positive predictions, while the recall is the number of true positive predictions divided by the total number of true-positives in the dataset. In the context of image comparison, precision and recall can be thought of as measures of how accurately the generated image matches the true image. The F1 score combines these measures into a single value, which provides an overall assessment of the similarity between the two images. Therefore, the use of the F1 score in this context is designed to quantitatively evaluate how well a generated image matches a true image with the same amount of activity (e.g. to determine whether or not a generated image with an X flare is similar to a true image with an X flare), based on the precision and recall of the classification model used to make the comparison. Thus, to evaluate how similar a generated image is to a ground-truth image of the same class, we train a supervised classifier on true data, test it on generated data looking at the F1 score per class, and then we take the macro F1 score.

6. Results

The results of these experiments are shown in Table 1. We produced a total of 60 000 images for these analyses, with each of the three models contributing 20 000 samples. Half of these images (30 000) were generated to reflect the proportions of image classes (A, B, C, M and X) found in the original dataset, while the other half (30 000) were generated uniformly with each image class equally represented. Specifically, for each of the three models, the generated images were separated into 20 sets of 1000 images each. In ten of these sets, each category (A, B, C, M, and X) is represented by 200 images (uniformly generated). In contrast, the class distribution in the remaining ten sets mirrors the imbalances evident in the original dataset, where each class is represented according to the percentages shown in Fig. 1. The reason for generating data with different distributions, as explained before, relies on the metrics used to evaluate the performance of our models. The cluster metrics and the FID indeed measure the similarity between the generated distribution and the true distribution, and so both of them need to be compared on a generated dataset with the same characteristics as the true dataset; in this case, in terms of class percentages. On the other hand, the F1 score, the precision and the recall are used to determine if a generated image of a particular class is similar to a true image of the same class, and therefore the trained classifier (see Sect. 5) should be tested on a uniform dataset without imbalance in the class percentages; otherwise our results would be biased.

Furthermore, for the cluster metrics we need a latent space to compute the calculations. For this reason, we decided to analyse different feature spaces using the t-SNE dimensionality-reduction technique (van der Maaten & Hinton 2008). The 512-dimensional ceVAE (Giger 2022) latent space is the most accurate representation of the class division (A, B, C, M, X); as shown in Fig. 3, it is the only latent space where clustering can be inferred, and the filamentous structure is related to the images that are close in time and thus very similar, underlying a major completeness with respect to the CLIP latent space and the classifier latent space. The classifier has been trained in a supervised manner, and the lack of distinct clusters in the latent space can be attributed to insufficient data in different classes. In such cases, it has been demonstrated that unsupervised methods can outperform supervised models (Voloshynovskiy et al. 2020).

The cluster metrics are based on the K-means unsupervised clustering with the Sklearn library (Buitinck et al. 2013). In Table 1, the results of the cluster metrics GEN (generated) should be compared to the values of the cluster metrics GT (ground truth), which serve as a benchmark. As there are five GOES classes A, B, C, M, and X, the number of clusters used for calculating these metrics is five. The model trained with discrete GOES classes has the lowest cluster error, while the model trained with continuous X-ray values has the lowest cluster distance and cluster standard deviation. The cluster error measures whether the clusters in feature space contain the same number of samples as the target distribution. Consequently, this metric has the potential to reveal mode collapse. This means that the model trained with the discrete labels can produce a distribution of data that can be clustered similarly to the true distribution. The cluster distance and standard deviation, on the other hand, determine whether the generated samples populate the correct regions in feature space with sufficient diversity and in this case the best is the model trained with x-ray continuous values. This suggest that the model trained with discrete labels is better at reproducing the overall structure of the data, while the model trained with continuous X-ray values is better at capturing the finer details and ensuring diversity in the generated samples. Nevertheless, as we can see from Fig. 4, the model trained with the GOES classes better differentiates between the energy classes. Indeed with the X-ray model will not be possible to generate images of the Sun in an extremely calm or extremely active

Table 1. Results of the experiments based on the metrics of Sect. 4.

Metric	ceVAE (baseline)	Discrete (ours)	Continous (ours)	ceVAE_Emb (ours)
Cluster error GT		0.00197		
Cluster distance GT		1.00104		
Cluster Std GT		0.99816		
Cluster error GEN ↓	7.9478 ± 0.9137	$\mathbf{0.1294 \pm 0.0358}$	1.5031 ± 0.1476	0.2073 ± 0.0361
Cluster distance GEN ↓	2.2057 ± 0.0096	0.9212 ± 0.0037	$\mathbf{0.9342 \pm 0.0023}$	0.8377 ± 0.0055
Cluster Std GEN ↓	3.2382 ± 0.0096	1.2107 ± 0.0037	$\mathbf{1.0976 \pm 0.0023}$	1.4801 ± 0.0055
FID CLIP ↓	5.05	0.122	**0.057**	0.39
FID IV3 ↓	215.933	3.693	**2.703**	12.264
F1 score ↑		**0.7**	0.34	0.6
Precision ↑		**0.73**	0.35	0.6
Recall ↑		**0.74**	0.37	0.7

Notes. The symbol ↓ indicates that a lower value is preferable for the metric it represents, while the symbol ↑ indicates that a higher value is preferable for that metric. The F1 score, precision, and recall are designed such that their maximum value is 1.

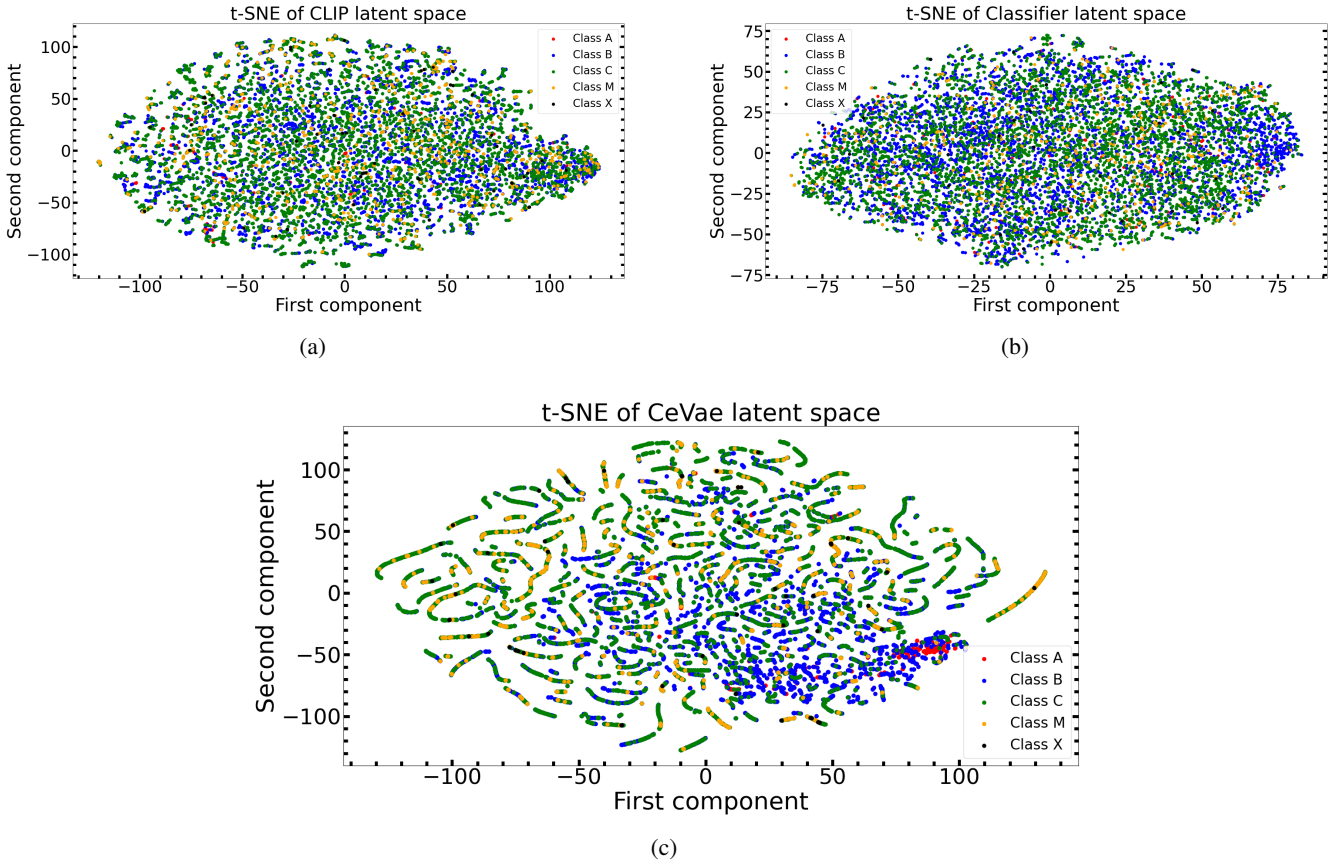

Fig. 3. t-SNE dimensionality-reduction technique applied to various latent spaces to determine which is most appropriate for cluster metrics. Panel a shows the t-SNE of CLIP latent space. Panel b shows the t-SNE of the latent space of a classifier. Panel c shows the latent space of a pretrained ceVAE.

state, resulting in a uniform production of activity across all levels.

For the FID, we used the Python library clean-fid (Parmar et al. 2021). The latent spaces considered for the FID are the ViT-B/32 CLIP encoder and the InceptionV3 encoder (Radford et al. 2021; Karras et al. 2020), specifically for consistency with the literature, so that we can make more meaningful comparisons of our results. More effective models are characterised by lower FID values, and here we therefore find the X-ray model to be the most effective. However, as seen in Fig. 4, the X-ray model is always generating activity and this leads to a lower value of the FID, as before for the cluster metrics. This means that the best is the model trained with the discrete GOES labels even though the FID is slightly higher with respect to the X-ray model. In addition to visual inspection, we can confirm this trend, with the F1 score, the precision and the recall at the end of Table 1 (these represent the macro values, which are the averages among the classes; the values for each class is given in Appendix D). As stated in Sect. 4, we trained a supervised classifier on the true data using the distilled data-efficient image

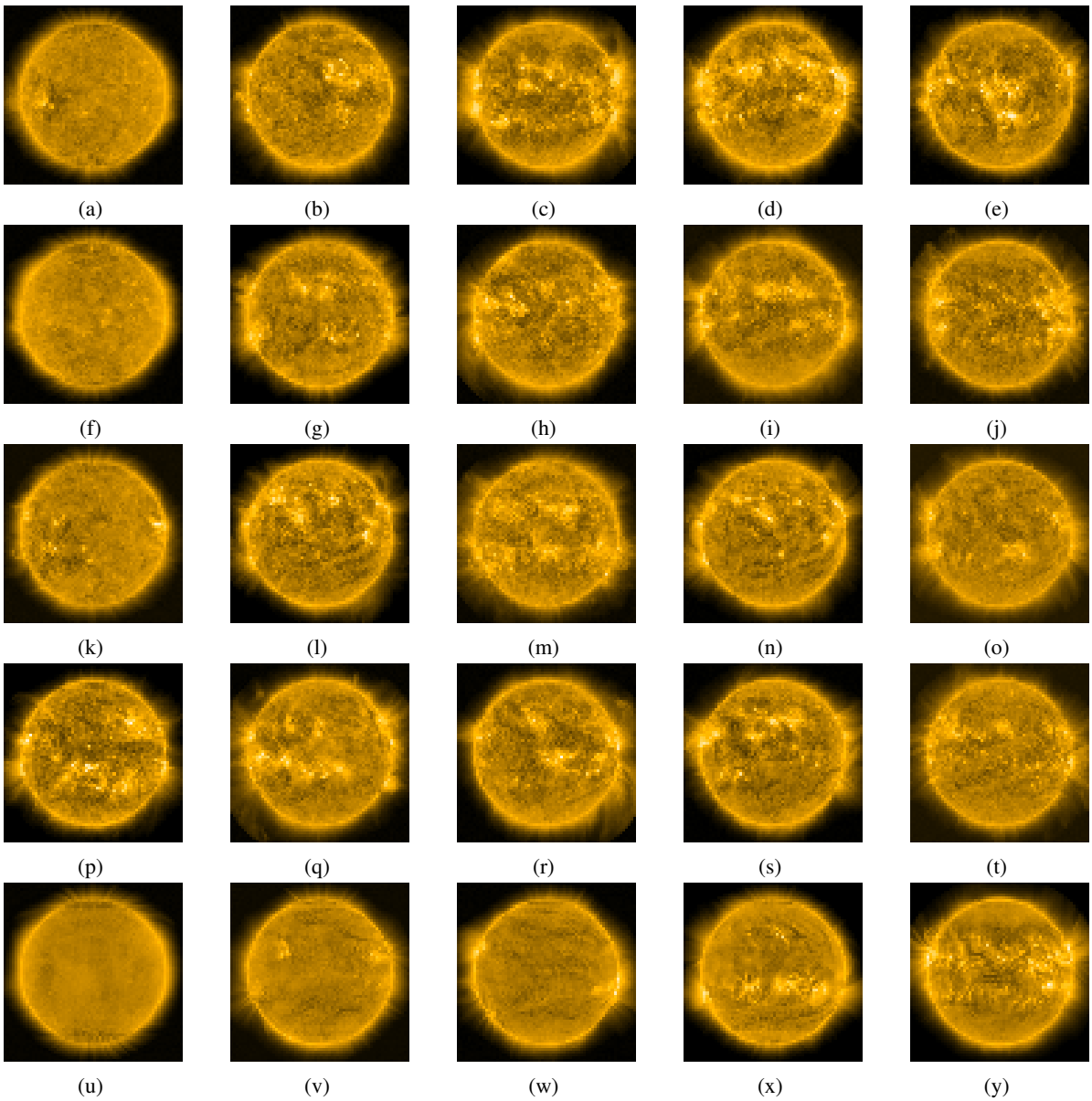

Fig. 4. Batch of 25 generated images. The first two rows are generated with the discrete label model, the third and the fourth row with the X-ray model and the last row with the ceVAE embedding model. The first column shows the A class, the second column the B class, the third column the C class, the fourth column the M class and the fifth column the X class.

transformer (DeIT) backbone (Touvron et al. 2020) to teach the model how to recognise true A-class, B-class, C-class, M-class image, and X-class images. The value of the F1 score, the precision and the recall on true data are respectively 0.55, 0.57 and 0.54. These benchmark values serve as a reference point. When assessing the performance of our trained classifier on the generated data, we compare the obtained results to those achieved on the true data by dividing the former by the latter. Performing this analysis on the ceVAE model is not feasible because it is not a conditioned model, and therefore, it is not possible to generate an image with a specific flare. Subsequently, we evaluate this classifier on generated data in order to determine which model produces images that are most similar to the actual images for the respective GOES classes. As a result of this analysis, the model trained with discrete GOES classes is the best model in terms of F1 score, precision, and accuracy, with a macro F1 score of 0.38, which is the 70% (0.7) of to the best score we can achieve (on true data) 0.54, whereas the X-ray model achieves only the 34% (0.34) as macro F1 score of the baseline.

7. Image analysis

To the best of our knowledge, our method is the first to generate images of the Sun with the ability to control its activity and the first to apply the novel concept of DDPM (Ho et al. 2020) to the field of heliophysics. Based on the results of Sect. 6, the best model in terms of visual inspection, distribution generation (cluster metrics) and applicability (F1 score) is the model trained with the discrete GOES labels: A, B, C, M and X. It is possible to control the presence and intensity of a solar flare on simulated SDO images of the Sun without copying the data from the training set, as evidenced by the fact that the cluster metrics do not perfectly match the reference values, and by the standard deviation maps (std maps) in Fig. 5. For every class

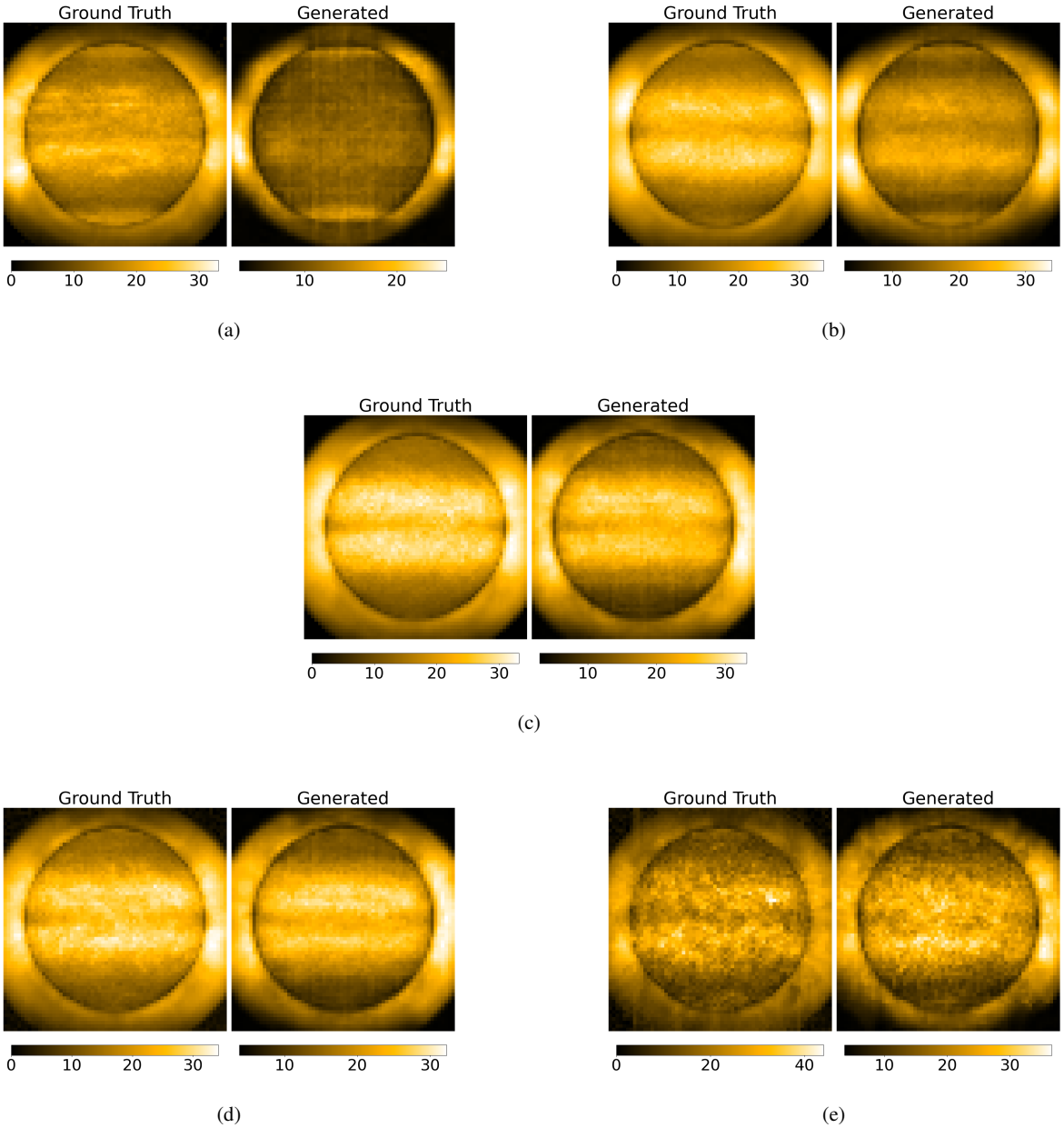

Fig. 5. Standard-deviation maps, comparing true images (left) and generated images (right) for each class. Panel a represents the A class, panel b the B class, panel c the C class, panel d the M class and panel e) the X class.

the images of the left panel in Fig. 5 represent the variation in the true images, whilst those in the right panel the generated images (using the discrete model). To compute the standard deviation maps, we concatenate the images along the batch dimension and then calculate the standard deviation per pixel, so that the brighter regions correspond to regions with greater variation and thus greater activity. We can see that the active regions on generated images are similar in terms of position to the real data but are never in exactly the same part of the image and with the same intensity, even for the X standard deviation maps with only 47 images in the training set. Furthermore, we never observe active regions at the Sun's poles, which is consistent with physical observations. However, the generated standard deviation maps on A images is the most divergent from the actual data. Indeed, the A-generated images are extremely stable, with few variations, regardless of the fact that the training set contains A images with some activity. Further tests are needed in order to better analyse this phenomenon (Somepalli et al. 2022) and we plan to carry on such tests in future works. Given the present findings, we can conclude that the model is able to generate all the types of activity present in the training set, with a minor limitations being its lack of ability to generate A-level images that are nearer to B-level than to low A-level.

8. Model usage

We now turn to two possible downstream experiments of the best model considered in this study based on the results obtained in Sect. 6. This procedure considers the strengths and weaknesses of the model, as well as its ability to generalise. Our ultimate objective is not for the model to beat all existing models on those tasks, but showing that the usage of generated images has a positive impact with respect to not using them.

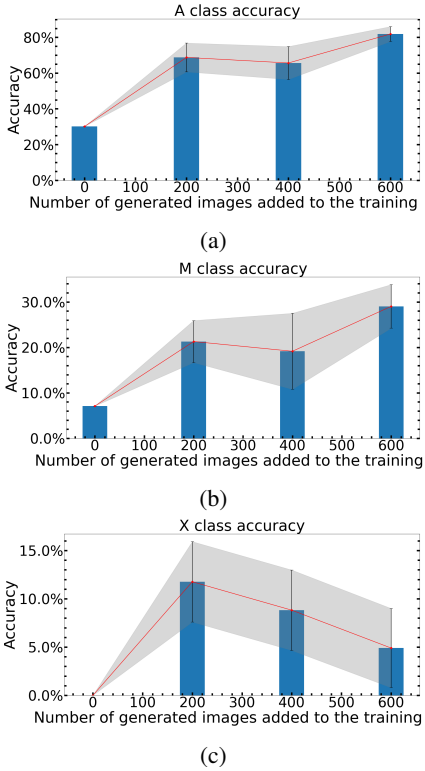

Fig. 6. Variation of accuracy per class increasing the number of generated samples added. Panel a shows the evolution of the accuracy of A class, panel b shows that of M class and panel c shows that of X class.

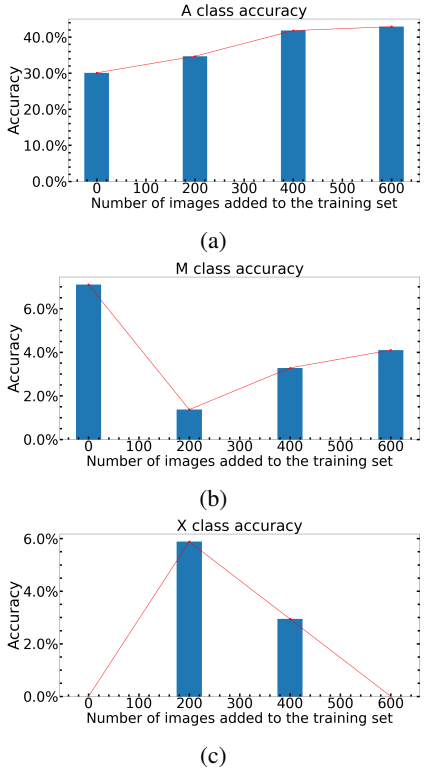

Fig. 7. Variation in accuracy per class, increasing the number of augmented samples added. Panel a shows the evolution of the accuracy of A class, panel b shows that of M class and panel c shows that of X class.

8.1. Classification experiment

Given the results in Table 1 and the potential of the DDPM to generate synthetic solar images, we tested whether or not we can use them to overcome the problem of unbalanced data. For this purpose, we trained a supervised classifier with the same architecture as in Sect. 4; we first did this without the addition of generated data to the least represented classes of the training set, A, M and X, and then we added 200, 400 and 600 synthetic images per represented class, respectively. The aim of this excercise is to see if adding the generated images improves the performance of our classifier, boosting the detection of underrepresented classes. In total, we trained four identical supervised classifiers, with the only difference between them being the addition of the generated samples. Naturally, each time, we tested on the same set of real data.

The proportion of added data is small compared to the size of the entire dataset. Therefore, the dataset remains unbalanced. This is intentional, as the experiment is to measure the impact of adding synthetic images – even in small numbers. For each incremental addition (200, 400, and 600), we used three distinct sets of generated images to better understand the resulting variations in the obtained values. In other words, for each addition, we trained three separate classifiers with different sets of 200, 400 and 600 images, respectively. This approach allows us to gain insight into the variations that arise from these different additions.

In Fig. 6, we can see that adding the generated data to the training set increases the accuracy for the three least represented classes, A, M, and X. The soft grey regions are the variation in the accuracy, and for the A and M classes, the more images we add, the better the detection; for example, for A class with 600 images we reach 81.9% accuracy compared to 30.1% without adding any synthetic data, and for M class with 600 images we reach 29.1% accuracy compared to 7.1%. The is not exactly the same for the X class. Figure 7c appears to show that adding a lot of data leads to a decrease in the accuracy gained. At this stage, we are not making any assumptions as to how the detection accuracy is going to develop by adding more data of a particular class. The literature (Yang et al. 2023) suggests that this depends on the initial size of the augmented classes, but also on the specific data used. We would need many further tests and more synthetic images to better explore this topic (e.g., adding 1000, 2000, 3000, or more images for all of the least represented classes).

As an additional test, we contrasted the outcomes achieved by incorporating generated images with those obtained using classical data-augmentation techniques applied to authentic images, employing the same methodology of involving three incremental stages (200, 400, and 600). We employed various data-augmentation techniques through a series of transformation compositions using the torchvision library (Paszke et al. 2019). The techniques used include: random horizontal and vertical flipping, random rotation with varying degrees, random affine transformations with specified degrees, translations, and shearing. As shown in Fig. 7, the application of classical augmented samples yields inconsistent results and does not consistently enhance performance. In certain instances, such as the M accuracy shown in Fig. 7b, there is a decline in performance compared to when no data-augmentation techniques are used.

We are aware that using all these classical data transformations could result in significant changes to the data distribution, but we take all of them in order to be able to compare with the

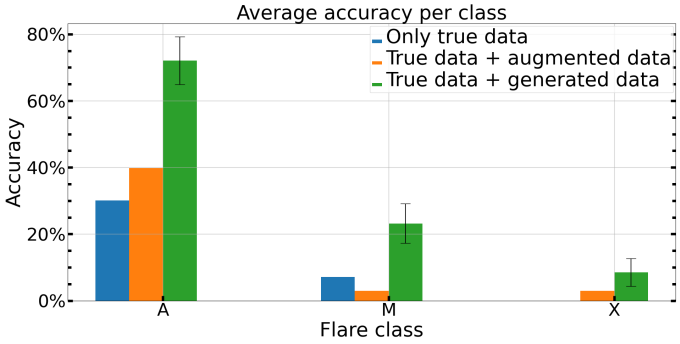

Fig. 8. Accuracy values of the least represented class with the addition of synthetic images, with only true images and with data augmentation on true images in the training set.

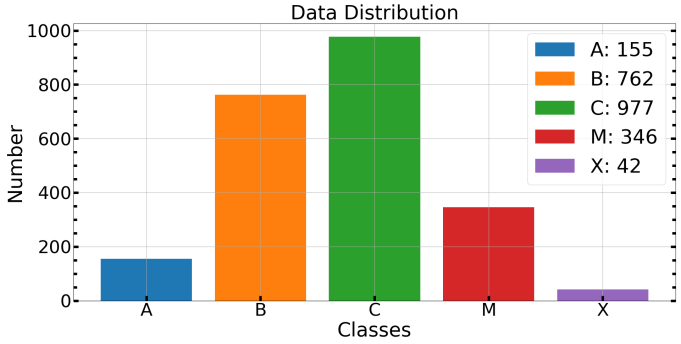

Fig. 9. Histogram distribution of the labelled dataset with the discrete GOES labels: A, B, C, M and X.

results obtained with the DDPM; otherwise using only vertical flipping or random rotation would not be enough for the lower represented class to produce 200, 400, 600 transformed images. In the case of the X class, our training set contains only 48 examples. This limitation makes it impossible to generate 200, 400, or 600 unique augmented data points without duplicating the same object, if we use only vertical flipping or random rotation. In the contrary using more transformations we avoid duplications but they lead to deviations from typical phenomena, leading to deteriorated results, as demonstrated in Fig. 7.

In contrast, augmentation using the diffusion model does not lead to this issue. With DDPM, we generate images that are not mere copies of the training data, as evidenced by the standard deviation maps in Fig. 5. These maps show, for instance, that there are no flare phenomena occurring at the poles. As we can see in Fig. 8, the addition of generated samples to the training set improves the performance of a supervised classifier by increasing its detection accuracy. Consequently, the technique utilised in this project is a valid method for overcoming the unbalanced dataset and for generating new images of the Sun in which we can control its level of activity.

8.2. Solar flare prediction experiment

Predicting solar flares is a critical task given the consequences outlined in Sect. 1. Generally, it is posed as a classification problem (Huang et al. 2018; Li et al. 2020; Pandey et al. 2023), where given input data $\mathbf{x}$ sampled at time t_0, the goal is to predict whether a flare will occur in the time window $t \in (t_0, t_0 + \Delta t]$, with Δt being arbitrarily chosen. There are various approaches to tackling this problem. For example, one approach is multi-class classification, the aim of which is to predict whether there will be an A, B, C, M, or X flare or their subclasses. Another approach is binary classification, where data are grouped based on the consequences of the solar flares; A, B, and C flares are grouped together, and M and X flares, which are more dangerous, are classified in another group. Further modelling criteria involve the use of either full-disc images (Pandey et al. 2023; Yi et al. 2023, as is done in these studies), or using patches to focus on the active regions (Zheng et al. 2019). From a machine learning perspective, using patches of the active regions as input can potentially enhance model performance per active region due to their high resolution. However, from an 'artificial intelligence' standpoint, one would expect the model to find out where to focus, eliminating the need for various preprocessing steps. Furthermore, it is possible to conduct a full-disc forecasts using a patch-based model and the output flare probabilities for each active region are typically aggregated. This approach treats all active regions independently and assigns them equal weight, which may not accurately reflect reality, as discussed in Pandey et al. (2022, 2023).

In this experiment, we conduct a full-disc solar flare prediction as a binary classification problem with a 24 h time window. In this setup, A, B, and C flares are categorised as class 0, while M and X flares are categorised as class 1. Our objective with this experiment is not to achieve state-of-the-art efficiency in solar flare prediction, but rather to demonstrate the impact of using images generated by the DDPM, as illustrated in Sect. 8.1.

However, the dataset described in Sect. 2.5 treats all flare events as independent samples – even if they occur from the same flaring region. For flare prediction, we need to be more selective and select only one image per subsequent 24 h window. For this reason we take a single image each day at 00:00:00 and label it as the most intensive flare that will occur in the next 24 h (e.g. if in the next 24 h there is a C and an X flare then, we label the image as X). After this preprocessing, we end up with a total of 2282 data points that follow the distribution shown in Fig. 9, which is the same as in Fig. 1. With this new dataset, we train a new conditional diffusion model with the same best setup found in Sect. 6 for generating the images. For the solar flare prediction architecture we keep the same DeIT backbone (Touvron et al. 2020), as described in Sect. 4. This architecture is combined with a weighted Cross-Entropy loss (assigning 0.1 to the majority class and 0.9 to the minority class) and employs a learning rate that decays following a cosine function. We train the model for 18 epochs using the same initial setup as previously described. This training is conducted under three different scenarios: without augmented data, with classical data augmentations, and with DDPM data-augmentations. The classical data augmentation techniques used include only vertical flipping and random rotation within a range of 10 degrees. The DDPM augmentations involve injecting varying amounts of data into the training set, ranging from 50 to 500 instances for the under-represented classes (M and X), which are both classified as 1 in our binary classification scenario. Notably, this augmentation does not include additional data from the more prevalent A, B, and C classes. The metrics taken into consideration are:

– The true skill statistics (TSS):

$$\mathrm{TSS} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} - \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}, \tag{7}$$

– the Heidke skill score (HSS):

$$\mathrm{HSS} = 2 \times \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FN} \times \mathrm{FP}}{(\mathrm{P} \times (\mathrm{FN} + \mathrm{TN}) + (\mathrm{TP} + \mathrm{FP}) \times \mathrm{N})}, \tag{8}$$

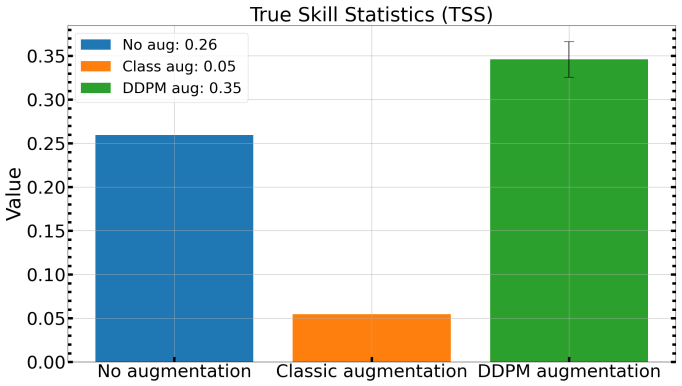

Fig. 10. True skill statistics values in the three different scenarios: without data augmentation, with classical data augmentation and with DDPM data augmentation.

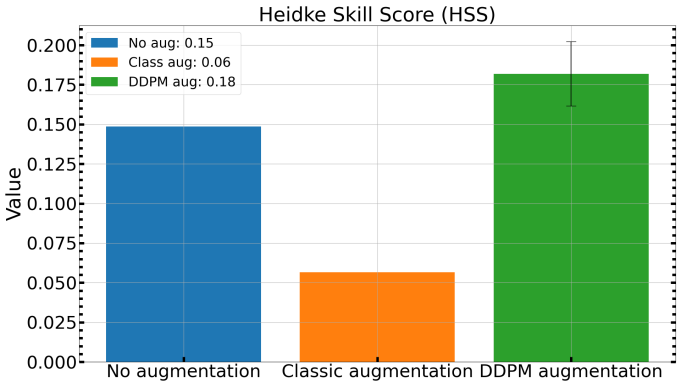

Fig. 11. Heidke skill score values in the three different scenarios: without data augmentation, with classical data augmentation and with DDPM data augmentation.

1. $N = TN + FP$ and $P = TP + FN$.

These metrics range from -1 to 1, where -1 indicates all incorrect predictions, 0 signifies performance equivalent to random guessing, and 1 denotes perfect predictions. Both metrics are employed in the context of solar-flare prediction because they are useful for assessing predictive performance, particularly in scenarios with imbalanced class distributions.

As observed in Figs. 10 and 11, employing DDPM augmentations consistently improves both the TSS and HSS metrics. The error bars represent the standard deviation calculated from using 50, 100, 200, 300, 400, and 500 generated data points per least-represented class in the training set, while the bars indicate the mean values. The final values, with DDPM-augmented data, are 0.35 ± 0.02 for the TSS and 0.18 ± 0.02 for the HSS. Classical data augmentation, instead, consistently decreases the performance, even when using data-augmentation techniques that should not deviate significantly from the normal distribution of the flares, such as vertical flipping and random rotation within 10 degrees. All the evaluation metrics are computed on the same test set, where there are only true data and no augmentations of any kind.

9. Conclusions

The goal of this work is to show the ability of the DDPM to generate images conditioned on the flare class so that they can be used in an equivalent way to the true images and thus prevent dataset imbalance towards the highest energy flares. It is possible to see from Fig. 3 – where different architectures are used to encode the image information from the 64×64 images to compute the metrics – that the ceVAE architecture presents some clustering and differentiation between the different flare classes with respect to the other architecture, and thus it is possible to highlight some differences between various classes even with a 64×64 image. Undoubtedly, in Fig. 3, the ceVAE latent space is not perfectly clustered and in a future work we will analyse the effects of increasing the image size and whether or not this will lead to a more definite clustering. The results are presented in Table 1.

In this Table, we trained a classifier on authentic data and subsequently evaluated on generated data. We find that the model successfully generates X-flare Sun instances that are very similar to authentic ones, despite the image dimensions being limited to 64×64. This suggests that the model effectively recognises the distinctions between various flare classes despite the limited image size. It is noteworthy that the average time interval between successive images in our dataset is 72 min. Consequently, certain images may exhibit minimal visual differences while being associated to distinct flare classes. The DDPM does not encounter any issues in this scenario, as it does not involve classification. Whenever this model is used, the related flare class is always provided as input with the image. Therefore, even if two images appear visually similar, the presence of the flare class serves as a discriminant. On the contrary, the application will not be able to distinguish all the data correctly, leading to greater uncertainty (Table 1). This uncertainty will be larger if we consider images that come from the same active region. The primary objective of this application is to demonstrate that by solely training a classifier without any fine tuning of the model, we were able to enhance performance in terms of the metrics employed here by using the synthetic images to balance the dataset. For this reason, we decided to subset the dataset in such a way that it is standardised for solar-flare prediction and to test the DPPM in this scenario. As is true for the classification task, in the solar-flare prediction task the use of the DDPM-augmented data improves the performance of the model when using the same setup as without data augmentation.

In future work, we would like to better comprehend the generation capabilities of the DDPM models (e.g. analysing the DDPM latent space), apply them to image-to-image translation tasks (Saharia et al. 2022, e.g. to obtain HMI magnetograms from each generated image), and to increase the image size to explore the impacts of this change. In addition, we would like to overcome the dataset limitations described in Sect. 2.5 and zoom in on the flaring regions, validating them with physical metrics so that they can be used for physics and machine learning-related downstream tasks. (Armstrong & Fletcher 2019; Love et al. 2020; Innocenti et al. 2021).

Acknowledgements. This research was partially funded by the SNF Sinergia project (CRSII5-193716): Robust Deep Density Models for High-Energy Particle Physics and Solar Flare Analysis (RODEM).

References

Armstrong, J. A., & Fletcher, L. 2019, Sol. Phys., 294, 80
Aschwanden, M. J., & Freeland, S. L. 2012, ApJ, 754, 112
Battaglia, A. F., Wang, W., Saqri, J., et al. 2023, A&A, 670, A56
Buitinck, L., Louppe, G., Blondel, M., et al. 2013, ECML PKDD Workshop: Languages for Data Mining and Machine Learning, 108

Chen, Y., Kempton, D. J., Ahmadzadeh, A., et al. 2022, Neural Comput. App., 34, 13339
Cicogna, D., Berrilli, F., Calchetti, D., et al. 2021, ApJ, 915, 38
Collier, H., Hayes, L. A., Battaglia, A. F., Harra, L. K., & Krucker, S. 2023, A&A, 671, A79
Dash, A., Ye, J., Wang, G., & Jin, H. 2022, Ann. Data Sci., https://doi.org/10.1007/s40745-022-00436-2
Dempster, A. P., Laird, N. M., & Rubin, D. B. 1977, J. R. Stat. Soc. Ser. B (Methodol.), 39, 1
Deng, J., Song, W., Liu, D., et al. 2021, ApJ, 923, 76
Dhariwal, P., & Nichol, A. 2021, CoRR, ArXiv e-prints [arXiv:2105.05233]
Fargion, D., Oliva, P., Lucentini, P. G. D. S., et al. 2019, Solar Neutrinos (World Scientific)
Galvez, R., Fouhey, D. F., Jin, M., et al. 2019, ApJS, 242, 7
Giger, M. 2022, Unsupervised Anomaly Detection with Variational Autoencoders in Heliophysics, https://github.com/i4Ds/sdo-cli
Gopalswamy, N., Xie, H., Yashiro, S., & Akiyama, S. 2023, ArXiv e-prints [arXiv:2303.02330]
Guastavino, S., Marchetti, F., Benvenuto, F., Campi, C., & Piana, M. 2023, Front. Astron. Space Sci., 9,1039805
Hackstein, S., Kinakh, V., Bailer, C., & Melchior, M. 2023, Astron. Comput., 42, 100685
Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. 2017, ArXiv e-prints [arXiv:1706.08500]
Ho, J., Jain, A., & Abbeel, P. 2020, ArXiv e-prints [arXiv:2006.11239]
Ho, J., & Salimans, T. 2022, ArXiv e-prints [arXiv:2207.12598]
Huang, X., Wang, H., Xu, L., et al. 2018, ApJ, 856, 7
Hurlburt, N., Cheung, M., Schrijver, C., et al. 2010, Sol. Phys., 275, 67
Huwyler, C., & Melchior, M. 2022, Astron. Comput., 41, 100668
Huy, P. N., & Quan, T. M. 2023, Arxiv e-prints [arXiv:2304.09383]
Innocenti, M. E., Amaya, J., Raeder, J., et al. 2021, Ann. Geophys., 39, 861
Karchev, K., Anau Montel, N., Coogan, A., & Weniger, C. 2022, ArXiv e-prints [arXiv:2211.04365]
Karras, T., Aittala, M., Hellsten, J., et al. 2020, ArXiv e-prints [arXiv:2006.06676]
Knipp, D. J., Ramsay, A. C., Beard, E. D., et al. 2016, Space Weather, 14, 614
Lemen, J. R., Title, A. M., Akin, D. J., et al. 2012, Sol. Phys., 275, 17
Li, X., Zheng, Y., Wang, X., & Wang, L. 2020, ApJ, 891, 10
Liu, A., & Carande, W. 2022, ESS Open Archive, https://doi.org/10.1002/essoar.10510080.1
Loshchilov, I., & Hutter, F. 2017, ArXiv e-prints [arXiv:1711.05101]
Love, T., Neukirch, T., & Parnell, C. E. 2020, Front. Astron. Space Sci., 7, 34
NOAA 2023, GOES Solar Flare Classification, https://www.swpc.noaa.gov/products/goes-x-ray-flux
Pandey, C., Ji, A., Angryk, R. A., Georgoulis, M. K., & Aydin, B. 2022, Front. Astron. Space Sci., 9, 897301
Pandey, C., Angryk, R. A., Georgoulis, M. K., & Aydin, B. 2023, Lect. Notes Comput. Sci., 14276, 567
Parmar, G., Zhang, R., & Zhu, J.-Y. 2021, ArXiv e-prints [arXiv:2104.11222]
Paszke, A., Gross, S., Massa, F., et al. 2019, ArXiv e-prints [arXiv:1912.01703]
Radford, A., Kim, J. W., Hallacy, C., et al. 2021, ArXiv e-prints [arXiv:2103.00020]
Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. 2022, ArXiv e-prints [arXiv:2204.06125]
Redmon, R. J., Seaton, D. B., Steenburgh, R., He, J., & Rodriguez, J. V. 2018, Space Weather Int. J. Res. App., 16, 1190
Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. 2021, ArXiv e-prints [arXiv:2112.10752]
Ronneberger, O., Fischer, P., & Brox, T. 2015, ArXiv e-prints [arXiv:1505.04597]
Saharia, C., Chan, W., Chang, H., et al. 2022, in Special Interest Group on Computer Graphics and Interactive Techniques Conference Proceedings (SIGGRAPH '22 Conference Proceedings) (New York, NY, USA: ACM), https://doi.org/10.1145/3528233.3530757
Sakurai, T. 2023, Physics, 5, 11
Salvatelli, V., dos Santos, L. F. G., Bose, S., et al. 2022, ApJ, 937, 100
Smith, D. S., & Scalo, J. M. 2007, Space Weather, 5, S06004
Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., & Ganguli, S. 2015, ArXiv e-prints [arXiv:1503.03585]
Somepalli, G., Singla, V., Goldblum, M., Geiping, J., & Goldstein, T., 2022 ArXiv e-prints [arXiv:2212.03860]
The SunPy Community (Barnes, W. T., et al.) 2020, ApJ, 890, 68
Tlatov, A.G., & Pevtsov, A.A. 2023, Geomagn. Aeron., 63, 863
Touvron, H., Cord, M., Douze, M., et al. 2020, ArXiv e-prints [arXiv:2012.12877]
Um, S., Lee, S., & Ye, J. C. 2023, ArXiv e-prints [arXiv:2301.12334]
van der Maaten, L., & Hinton, G. 2008, J. Mach. Learn. Res., 9, 2579
Vaswani, A., Shazeer, N., Parmar, N., et al. 2017, ArXiv e-prints [arXiv:1706.03762]
Voloshynovskiy, S., Taran, O., Kondah, M., Holotyak, T., & Rezende, D. 2020, Entropy, 22, 943
Wan, J., Fu, J.-F., Liu, J.-F., et al. 2021, Res. Astron. Astrophys., 21, 237
Wolleb, J., Bieder, F., Sandkühler, R., & Cattin, P. C. 2022, ArXiv e-prints [arXiv:2203.04306]
Xu, X. H., Wang, Y., Wei, F. S., et al. 2023, Sci. Rep., 13, 6101
Yang, Z., Shao, J., & Yang, Y. 2023, Big Data Res., 34, 100409
Yi, K., Moon, Y.-J., & Jeong, H.-J. 2023, ApJS, 265, 34
Zheng, Y., Li, X., & Wang, X. 2019, ApJ, 885, 73
Zimmerer, D., Kohl, S. A. A., Petersen, J., Isensee, F., & Maier-Hein, K. H. 2018, ArXiv e-prints [arXiv:1812.05941]
Zimmermann, R. S., Schott, L., Song, Y., Dunn, B. A., & Klindt, D. A. 2021, ArXiv e-prints [arXiv:2110.00473]

Appendix A: Dataset limitation

As introduced in Sect. 2.4, the limitation of our dataset is related to the time resolution of the AIA instrument in the SDOMLv2, which is 6 minutes instead of 12 seconds as in the original SDO data. As mentioned by Galvez et al. 2019, this is done in order to perform the temporal synchronisation with the EVE instrument. This is a limitation due to the fact that we are interested in a specific time when searching for the SDO image, as we only want to consider flaring occurrences. In fact, when cross-correlating the HEK dataset with the SDOMLv2 dataset, we use a time tolerance of 7 minutes to maximise the number of images, while bearing in mind that we may lose flaring information if the closest image is more than 10 minutes away in time. In Figure A.1, the time delays per class are depicted.

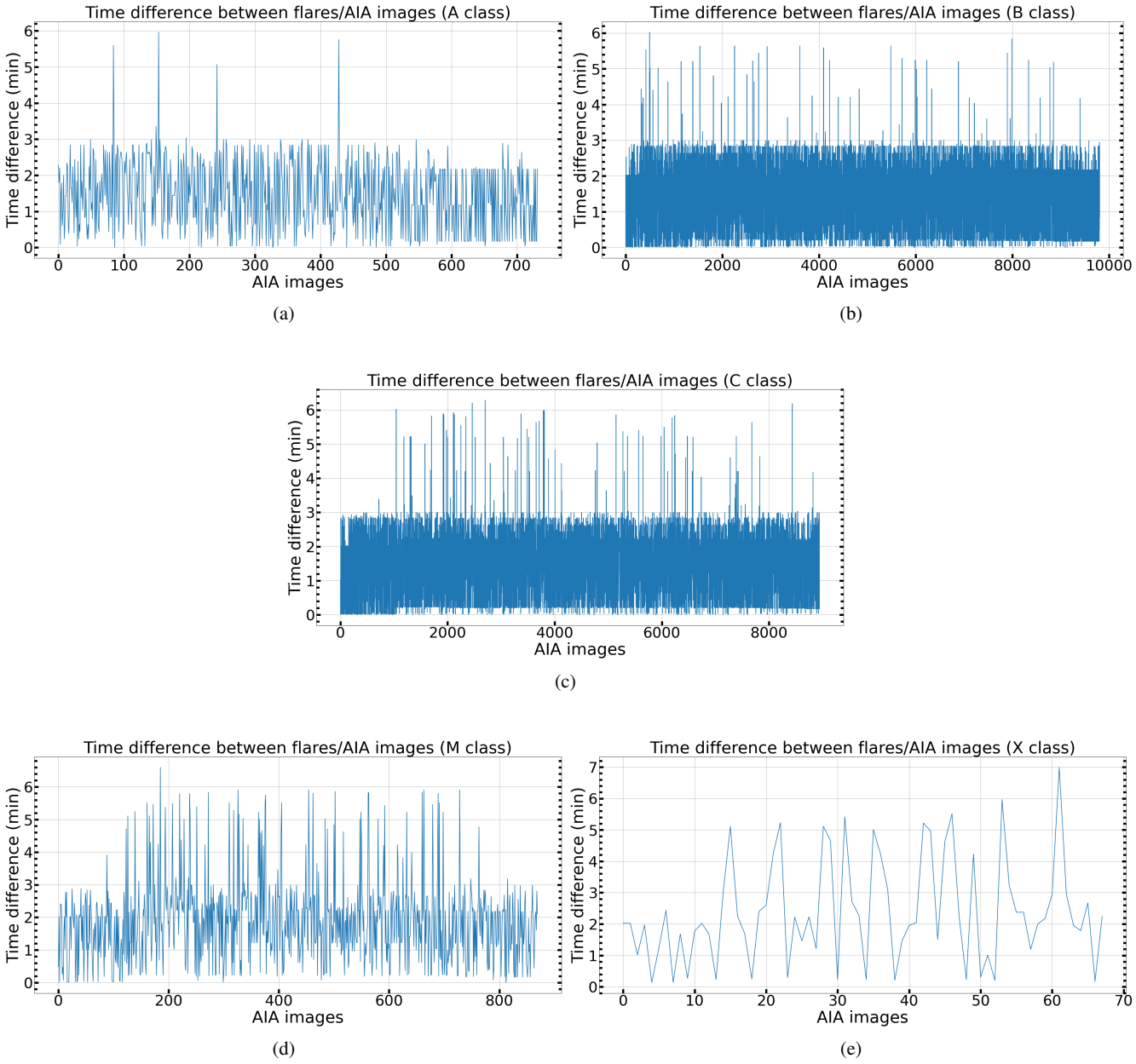

Fig. A.1. Time-delay histograms per flare class of the AIA image with respect to the peak time of the flaring event. Panels a) to e) represent the time delay of the images belonging to classes A to X, respectively.

Appendix B: Architecture

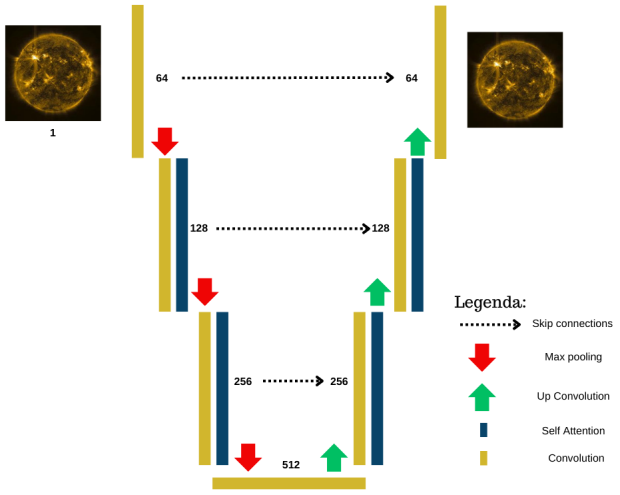

Fig. B.1. Unet architecture implementation.

The U-Net architecture is a form of convolutional neural network (CNN) that was initially developed for biomedical-image segmentation, but has since been applied to other image-segmentation issues as well. The network is known as U-Net because its architecture is U-shaped, with a contracting path (encoder) on the left and an expanding path (decoder) on the right. The contracting path is comprised of convolutional and pooling layers that gradually decrease the spatial resolution of the input image, whereas the expanding path employs upsampling and convolutional layers to gradually increase the resolution and generate a segmentation mask. In addition, U-Net includes skip connections that directly link the layers between the encoder and decoder channels. These skip connections enable the network to propagate information from the contracting path to the expanding path at varying spatial resolutions, thereby preserving high-resolution characteristics. In conclusion, our implementation (Figure B.1) between every downsampling and upsampling layer, includes a self-attention layer (Vaswani et al. 2017), which is used to model long-range dependencies between different spatial locations in an image. In this instance, the self-attention mechanism computes the relative importance of each spatial location in an image relative to other spatial locations. This is accomplished by applying a set of learned weight vectors to the input feature map to generate a set of attention maps that indicate the importance of each spatial location. The attention maps are then used to re-weight the input feature map, emphasising the most significant spatial locations and omitting the less significant ones. This generates a new feature map that contains the most pertinent data for the image-generation assignment.

Appendix C: Image generation with 128x128 pixel resolution

Denoising diffusion probabilistic models are very computationally expensive, but are very good in manipulating the details (Dhariwal & Nichol 2021). Indeed, increasing the resolution from 64x64 to 128x128, we can see (Figure C.1) that the generated images do not introduce physical artefacts at first sight, but further analysis should be carried in this regard. On the other hand, to be able to perform this generation, we cannot use the NVIDIA TITAN X GPU due to vram shortage (12 GB), but we use the NVIDIA A100 GPU with a vram of 40 GB. Despite this, we decrease the complexity of our architecture, removing two self-attention layers in the U-Net.

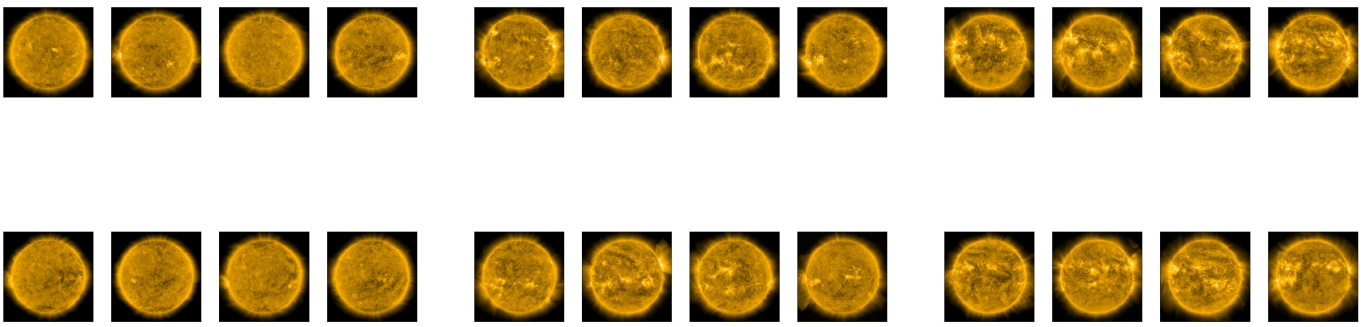

Fig. C.1. Generated images with 128x218 pixel resolution. Low level refers to A-class flares, medium level to B-class flares and high level to C-, M- and X-class flares. Panel a) shows the low-level activity, b) medium-level activity and c) the high-level activity.

Appendix D: F1-score, precision and recall

Table D.1. Metric results of classifier trained and tested on true data.

Class	F1-score	Precision	Recall
A	0.80	0.76	0.83
B	0.26	0.50	0.17
C	0.55	0.54	0.56
M	0.70	0.74	0.65
X	0.42	0.35	0.51

Table D.2. Metric results of the classifier trained and tested on generated data from the model with discrete GOES labels.

Class	F1-score	Precision	Recall
A	0.77	0.70	0.85
B	0.24	0.40	0.17
C	0.28	0.23	0.35
M	0.37	0.30	0.47
X	0.26	0.46	0.18

Table D.3. Metric results of the classifier trained and tested on generated data from the model with discrete GOES labels and the ceVAE embeddings.

Class	F1-score	Precision	Recall
A	0.67	0.58	0.79
B	0.23	0.34	0.18
C	0.32	0.27	0.39
M	0.45	0.38	0.56
X	0.04	0.13	0.02

Table D.4. Metric results of the classifier trained and tested on generated data from the model with the xray values.

Class	F1-score	Precision	Recall
A	0.19	0.21	0.17
B	0.16	0.19	0.14
C	0.24	0.20	0.32
M	0.23	0.19	0.27
X	0.12	0.19	0.09