Machine Learning: Science and Technology

PAPER • OPEN ACCESS

Data-driven plasma modelling: surrogate collisional radiative models of fluorocarbon plasmas from deep generative autoencoders

To cite this article: G A Daly *et al* 2023 *Mach. Learn.: Sci. Technol.* **4** 035035

View the article online for updates and enhancements.

You may also like

- The LHC Olympics 2020 a community challenge for anomaly detection in high energy physics
  Gregor Kasieczka, Benjamin Nachman, David Shih et al.

- Interpretable embeddings from molecular simulations using Gaussian mixture variational autoencoders
  Yasemin Bozkurt Varolgüne, Tristan Bereau and Joseph F Rudzinski

- Deep learning in electron microscopy
  Jeffrey M Ede

This content was downloaded from IP address 144.173.255.206 on 24/01/2024 at 14:51

IOP Publishing Mach. Learn.: Sci. Technol. 4 (2023) 035035 https://doi.org/10.1088/2632-2153/aced7f

MACHINE
LEARNING
Science and Technology

CrossMark

OPEN ACCESS

RECEIVED
12 April 2023

REVISED
6 July 2023

ACCEPTED FOR PUBLICATION
4 August 2023

PUBLISHED
29 August 2023

Original Content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence.

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

PAPER

Data-driven plasma modelling: surrogate collisional radiative models of fluorocarbon plasmas from deep generative autoencoders

G A Daly[1,2,*], J E Fieldsend[1], G Hassall[2] and G R Tabor[1]

[1] Faculty of Environment, Science and Economy, University of Exeter, North Park Road, Exeter EX4 4QF, United Kingdom
[2] Oxford Instruments Plasma Technology, North End, Yatton BS49 4AP, United Kingdom
[*] Author to whom any correspondence should be addressed.

E-mail: gd351@exeter.ac.uk

Keywords: plasma, autoencoder, generative, surrogate model, deep learning, semiconductor

Abstract

We have developed a deep generative model that can produce accurate optical emission spectra and colour images of an ICP plasma using only the applied coil power, electrode power, pressure and gas flows as inputs—essentially an empirical surrogate collisional radiative model. An autoencoder was trained on a dataset of 812 500 image/spectra pairs in argon, oxygen, Ar/O_2, CF_4/O_2 and SF_6/O_2 plasmas in an industrial plasma etch tool, taken across the entire operating space of the tool. The autoencoder learns to encode the input data into a compressed latent representation and then decode it back to a reconstruction of the data. We learn to map the plasma tool's inputs to the latent space and use the decoder to create a generative model. The model is very fast, taking just over 10 s to generate 10 000 measurements on a single GPU. This type of model can become a building block for a wide range of experiments and simulations. To aid this, we have released the underlying dataset of 812 500 image/spectra pairs used to train the model, the trained models and the model code for the community to accelerate the development and use of this exciting area of deep learning. Anyone can try the model, for free, on Google Colab.

1. Introduction

Generative models are a type of deep learning (DL) model that can produce new, unseen samples when trained on un-labelled data. These types of models have not been used previously in the field of low-temperature plasmas, but have been used to great effect in generating text, images and 3D models. They can offer many benefits by creating synthetic data for modelling and experiment design, replacing parts of computational models with fast surrogate models and providing a foundation for models that predict expensive and difficult to measure parameters from simpler diagnostics.

1.1. Background

Synthetic data can be an extremely useful resource in plasma physics for developing experiments, understanding diagnostics and training models and controllers for plasma applications. Synthetic data tools have been used in fusion [1–4] and laser plasmas [5–9] to aid simulations, experiment design and for training machine learning (ML) and DL models. However, such approaches have been used less frequently in low-temperature plasmas [10–12].

Methods for generating synthetic data, used in plasma physics, can be split into three main groups—generating synthetic sensor data from simulation or analytic models [1, 4–9, 12], inverting analytic methods for extracting parameters from sensor data [2, 10, 11] and augmenting existing experimental data to create new data [3]. However, DL generative models have not been used for synthetic data generation in plasma physics. This approach uses DL models, such as autoencoders (AEs), generative adversarial networks, diffusion models or transformers as a generative model that can create new synthetic data (see [13] for a recent review of the area). Outside the field these approaches have been used for improving medical image

© 2023 The Author(s). Published by IOP Publishing Ltd

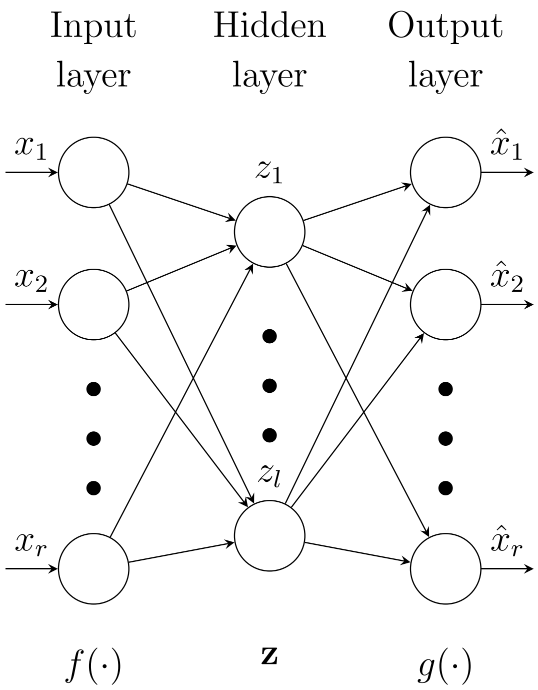


Figure 1. Schematic diagram of a basic autocencoder.

classification [14], drug design [15], chemical reaction discovery [16], cyber security [17], music generation [18] and image generation [19], and many other applications besides.

DL approaches have had many successes in the field, applied to controlling atmospheric pressure plasma jets [20, 21], a fast replacement for computed tomography for tokamak radiation profiles [22], classifying particle defects on semiconductor wafers [23–25], predicting electron energy distribution functions from optical emission spectra (OES) [26] and creating surrogate models of neutral beam injection [27], sputtering processes [12] and plasma etching [28].

In this work we demonstrate how deep AEs can be used to generate synthetic sensor data from large amounts of unlabelled experimental data. We show how to train a deep AE on unlabelled data and then how to train a model to learn to 'map' from an input space of physical variables into the latent space of the AE to produce a generative model.

In the context of the literature on DL, there has been a great deal of interest in developing generative models for some time, such as variational AEs (VAE) [29], generative adversarial models [30] and diffusion models [31]. Earlier work focused on developing models that were capable of generating good outputs through random sampling, more recent work has focused on how to guide generative models to produce desired generative outputs. This can be referred to as learning a prompt for generative output or a map to a latent space. Recent examples include generating music [18, 32], transforming facial expressions [33] generating energy angle distributions in sputtering processes [12], and new high quality image generation from prompt models such as DALLE 2, parti and stable diffusion [19, 34, 35].

1.2. AEs

AEs are an early type of neural network model that learns to copy its input at its output [36]. AEs consist of an encoder, $\mathbf{z} = f(\mathbf{x})$, that learns to map input data, $\mathbf{x} \in \mathbb{R}^r$, into a latent space ($\mathbf{z} \in \mathbb{R}^l$) and a decoder, $\hat{\mathbf{x}} = g(\mathbf{z})$ that learns to map the latent space representation back to the input [36], see figure 1. The model is trained to minimise the reconstruction error between the input data and the reconstructed output. On the face of it this does not seem like a very useful network, but by making the latent space much smaller than the input data ($l << r$), the network is forced to learn a low dimensional representation of the input data by learning relationships and patterns within the input data.

VAEs are an extension of ordinary AEs, where an additional training objective, the Kullback–Leibler (KL) divergence, is added to guide the distribution of the latent space to follow a normal distribution with a diagonal covariance matrix, $\mathbf{z} = \mathcal{N}(\mathbf{z}; 0, \mathbf{I})$. This gives VAEs a continuous latent space that can be easily sampled from to generate new samples. This has lead to VAEs being widely used in the field of generative modelling, however, they have had issues from their inception, as they are difficult to train and suffer from mode collapse [37, 38] and that the latent space does not always end up having the desired property of being a normal distribution [39], such as in figure 4 of [12]. The VAE prior itself has also been highlighted as a source of many of these problems due to over-regularisation creating uninformative latent representations [40, 41].

Recent work in the field of generative modelling has demonstrated that the VAE process can actually hamper the ability of the model to learn a useful representation through over-regularisation and that large AEs are good generative models, outperforming VAEs repeatedly [42–44]. In recent work, AEs have been used to learn features for virtual metrology models from optical emission spectroscopy (OES) [45] and defect detection in semiconductor processing [46]. We use AEs in this work as they are easier and more predictable to train than VAEs, while providing equal or better performance as a generative model, making them more suitable for widespread use in scientific applications.

Our contributions in this work and the structure of the paper are laid out as follows. In section 1 we provide a background to synthetic data generation, deep generative models and how it has been applied in other fields. In section 2 we describe how we created an experiment to gather 812 500 OES and colour images in fluorocarbon plasmas in an industrial plasma etcher. In section 3 we describe how to build and train an AE and how to train a small model to map physical tool inputs to the latent space and turn the decoder into a conditional generative model. In sections 4 and 5 we look at the structure of the latent space produced by the model for different sizes of latent space and the difficulty of evaluating generative models. In section 6 we demonstrate using the generative model to carry out synthetic experiments looking at line ratios in Argon and Ar/O_2 plasmas covering 10 000 points varying power and pressure in seconds. We consider any limitations of the approach and future work, and detail the open source release of code and experimental results in sections 7 and 8, followed by a conclusion to the work in section 9.

The data set we have gathered has been released under a creative commons license (CC BY-4.0) and can be used by anyone for academic purposes. The model's code and pre-trained models have been released as open source under the MIT License.

2. Data collection and experimental design

A dataset of 812 500 OES and RGB images of the bulk plasma above the wafer surface were gathered from an Oxford Instruments Plasma Technology PP 100 industrial plasma etcher with a Cobra300 cylindrical ICP source. Quartz windows were used for all optical diagnostics, for OES an Edmund Optics UV/VIS collimator (88–173) was used to collect light into a Thorlabs round to linear fibre bundle, consisting of seven 200 μm solarisation resistant fibres. An Avantes ULS4096CL-EVO-RM 200–1100 nm spectrometer was used with a 10 μm slit. Optical images were collected with a FLIR Blackfly 0.4 MP colour camera (BFS-U3-04S2M-CS) and a 6 mm focal length lens (SV-0614 V).

Data was collected across the entire operating region of the plasma source in argon, oxygen, Ar/O_2, CF_4/O_2 and SF_6/O_2. The experimental operating space consisted of the power delivered to the ICP source, the power to the table, the pressure in the chamber and the flow rate of one or two gases. The operating space varied for each gas due to differing lower limits on the minimum power and pressure to form a stable plasma or the requirement to keep the DC bias below 1 kV. The operating space is summarised in table 1.

Our aim was to make measurements at sample points across the operating space and gather the most amount of information within a fixed budget of samples. Naively, we could have used a grid search, however, a 10 point grid across five dimensions would require 100 000 points with very poor space filling, i.e there would be only ten unique values in each dimension. The next simplest approach would be to sample randomly, for large numbers of samples—this is quite likely to fill the parameter space, but there is no guarantee on how efficiently we can fill the operating space. The efficiency of filling a space and how well the points are separated can be measured by the discrepancy of the entire set, in particular, we use the L2 discrepancy to measure this [47, 48].

Quasi-random sequences offer a very effective way to generate sets of sample points that offer some guarantees on efficiency of filling a parameter space while still providing enough random spread to cover the interactions of many variables [47, 48], i.e. they have a low discrepancy, as shown in table 2. Two of the most common quasi-random sequences are Latin hypercube sampling (LHS) and Sobol sequences, both have the properties that we desire, but Sobol sequences have an advantage the you can generate further elements of the sequence, using the same random seed. This is important if you need to extend your dataset at a later time point. There is no guarantee that the combination of two LHS sets does not have a higher discrepancy than one generated with the combined number of data points and you cannot truncate or randomly sample from a large LHS and maintain the low discrepancy. However, with a Sobol sequence you have a guarantee that the extension to your dataset has the same discrepancy as if you had started by generating the sequence of that length [47, 49].

Using a Sobol sequence, we generated 10 000 points each for argon and oxygen, 30 000 points for Ar/O_2 and 60 000 for CF_4/O_2 and 70 000 for SF_6/O_2. To actually cover the entire sequence in our experiment, we

Table 1. Dataset setpoints.

	Argon	Oxygen	Ar/O_2	CF_4/O_2	SF_6/O_2
ICP/W	480→3000	600→3000	750→3000	600→3000	750→3000
Table/W	0→600	30→600	30→540	30→600	30→600
Pressure/mT	5→90	5→90	5→80	4→90	5→80
1st gas/sccm	3.5→70	2.5→50	2.5→50	4.2→84	2.6→52
2nd gas/sccm			2.5→50	2.5→50	2.5→50
Number of points, *n*	10 000	10 000	30 000	60 000	70 000

Table 2. L2 discrepancy of different sampling methods in five dimensions (lower is better, bold is best).

No. points	Grid	Random	Sobol
10^3	1.14×10^{-1}	1.54×10^{-3}	$\mathbf{2.52 \times 10^{-5}}$
10^4	2.87×10^{-2}	1.08×10^{-4}	$\mathbf{1.83 \times 10^{-7}}$
10^5	1.38×10^{-2}	1.80×10^{-5}	$\mathbf{4.63 \times 10^{-9}}$
10^6	5.09×10^{-3}	1.28×10^{-6}	$\mathbf{1.03 \times 10^{-10}}$

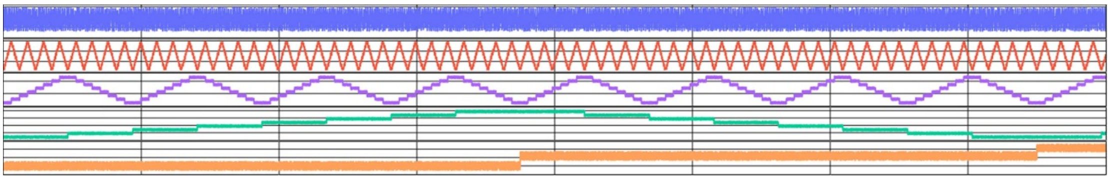


Figure 2. Ordered sweep from Sobol sequence for CF_4/O_2, from top to bottom—IPC power, table power, O_2 flow, CF_4 flow, pressure.

Table 3. Raw measured points.

	Argon	Oxygen	Ar/O_2	CF_4/O_2	SF_6/O_2
ICP/W	0→2997	0→2996	0→2997	68.1→2996	0→2996
ICP 0.1%→99.9%	464→2985	544→2988	72.2→2988	224→2988	595→2988
Table/W	0→613	0→604	0→544	2.75→614	0→545
Table 0.1%→99.9%	0.2→597	19.7→598	8.9→537	83.2→597	6.6→535
Pressure/mT	5→92	5→91	2.8→91	3.8→85.8	4.3→82.1
1st gas/sccm	3.5→70	0.1→50	2.8→70	4.2→84	0→52
2nd gas/sccm			2.5→50	2.5→50	0→50
Number of points	50 000	50 000	150 000	225 000	337 500

sorted each sequence such that pressure followed a relatively flat ramp over the whole range and other variables followed a triangle wave shape of increasing speed, as shown in figure 2. This enabled us to maintain tool stability between sample points and reduced the settling time between setpoint changes. Setpoints were changed every 5 s and a optical image and OES were taken every second starting at the beginning of the setpoint change. A plain, un-patterned, silicon wafer was clamped to the table at all times and the process was only stopped to replace the wafer when it had become too thin from etching.

The dataset consists of five image spectra pairs, $[i_{n,0},\ldots,i_{n,4}]$, $[s_{n,0},\ldots,s_{n,4}]$ and setpoint readbacks from the tool $[t_{n,0},\ldots,t_{n,4}]$, taken at each setpoint $[P_0,\ldots,P_n]$ for each gas mixture. The setpoint readbacks consist of the net power (forward-reflected) on the ICP coil and table, pressure in the chamber, gas flow from each mass flow controller and DC bias at the table.

The experimental points sampled did not perfectly align with our planned sweeps; some areas had unstable plasmas, could not sustain a plasma or exceeded parts of the tool's operational envelope, such as pressure control. The measured data is summarised in table 3, all of the runs have a small portion of results with momentary high reflected power, but not for long enough to cause the plasma to extinguish. In CF_4/O_2 plasma the high pressure region above 70 mT was unstable due to a combination of reduced plasma stability and limited control margin of the pressure controller and the sweeps were not continued above this pressure. In SF_6/O_2, the minimum power required to sustain a plasma increased with pressure and so the sequence was extended to 70 000 points and the minimum ICP power raised to 1500 W above 40 mT to yield more measurement points. The experiment yielded a total of 812 500 image spectra pairs, at 162 500 unique setpoints in the operational space of the tool.

The data was split into train, validation and test sets with a 80/10/10 split. However, since we hold and take five measurements at each set point, naively randomly splitting the data would result in leakage from the test data into the train split, i.e. some measurements at a single setpoint would be present in each split. To avoid this, the data is kept together in blocks of 5 and the blocks are randomly assigned to the three sets. The spectra are processed by subtracting the average of the counts at the dark pixels from each spectra and removing the data from pixels outside the calibrated range of the spectrometer, this leaves 3072 pixels covering 200–1100 nm. The intensity of each spectra is min-max scaled to between 0 and 1 and a 5 pixel wide Hann window [50] is used to smooth out noise in the spectra. The camera produces a 720 × 540 pixel image with an RGGB Bayer mask, rather than perform standard Bayer interpolation to produce a 720 × 540 colour image, we treat the camera like a hyperspectral camera with very poor spectral resolution. We take all the red and blue pixels and one of the green pixels to form three 360 × 270 images. These are cropped to the central area of the image, resized and stacked to produce a 128 × 96 × 3 image. The pixel intensities are well controlled by the camera's autoexposure algorithm and are all clustered around a 50% grey value, requiring no further normalisation. The camera ADC is set to a 10-bit resolution and values are stored as 16-bit integers, all images are divided by 2^{16} to rescale their pixel intensities between 0 and 1. The values from the tool's setpoint readbacks are all in the range of 0–10 V or 0–5 V and are simply divided by 10 to rescale them between 0 and 1.

This process of the rescaling and normalisation of inputs is a particularly important step in preparing data for training in any ML approach. It speeds up and stabilises convergence in training the model [51, 52], as gradients in the model will be within expected bounds for the optimiser and the inputs are within the expected bounds of activation functions, such as sigmoid and ReLU.

3. Building deep generative AEs for synthetic data generation

Our model architecture is based on ConvNeXt, a state of the art convolutional neural network architecture [53]. We use the base ConvNeXt blocks and stem, with 1D or 2D convolutions for OES or images to form our image and spectra encoding branches, the basic block is shown in figure 3. Each branch consists of four stages with (2, 2, 6, 2) blocks and (64, 128, 256, 512) filters, at the beginning of each stage a convolutional downscaling halves the spatial dimensions of the image or spectra. At the end of the last stage a global average pooling layer reduces all of the spatial dimensions and produces a single tensor with the size of the last set of filters and this is followed by two densely connected neural network layers of 1024 neurons and the chosen size of our latent space. The latent output of each branch is then summed together producing a tensor with the length of the latent space dimension and finishes in a dense layer with $\mathbf{z}$ neurons with a linear activation function. This is our latent representation of the input data and can be the combination of any number of input branches. The model was trained on different sized latent spaces, $l = [4, 16, 32, 64, 128]$, to demonstrate the effect the size of the latent space has on the model.

In this work we have only used two branches, both based on convolutional networks, but any number of branches can be used with any kind of network architecture encoding some input data. The decoder is simply the reverse of the encoder and finishes in a 1D or 2D convolution that reconstructs the input.

The encoder learns a function to project the input image and spectra i_n, s_n pair into a latent space, $z_n = f(i_n, s_n)$, each decoder branch then learns a function to project the latent space vector back into the real diagnostic space, $\hat{i_n} = g(z_n)$, $\hat{s_n} = h(z_n)$, this overall structure is shown in figure 4. The loss is a reconstruction loss between input, i_n, s_n, and reconstructions, $\hat{i_n}, \hat{s_n}$. This loss can be weighted to favour one input over another to embed prior assumptions about the relative importance of each diagnostic.

The model is trained with the Adam optimiser [54], using a cosine decay learning rate schedule [55] with a linear warmup, and mean-squared error (MSE) as the loss, using Keras [56]/Tensorflow [57]. Full details of the training and fine-tuning settings are in table 4. The model was trained on 4 Nvidia A100 GPUs for 100 epochs, taking roughly 20.5 h to train.

3.1. Tool to latent model architecture

Our decoder model can be used on its own for generative modelling, by randomly sampling over values of $\mathbf{z}$ we can generate random output spectra and images from our model, however, this is of limited practical use. To make this model into a synthetic data generator we need an additional model to learn to map from tool parameters $\mathbf{t}$ to the latent space, $\mathbf{z} = f(\mathbf{t})$. This is similar in its way of thinking to text-to-image models, such as stable diffusion [35], where the model is trained with pairs of text descriptions and images. In this work we train an additional model to produce latent representations, $\mathbf{z}$, from tool parameters that match the ones from their associated image and spectra pair. The parameters used were the net power on the ICP coil, table power, gas flows and pressure.

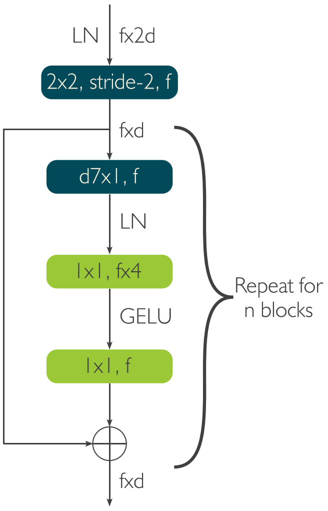


Figure 3. 1D Convnext basic block with layer normalisation (LN) and Gaussian error linear unit (GELU) activations.

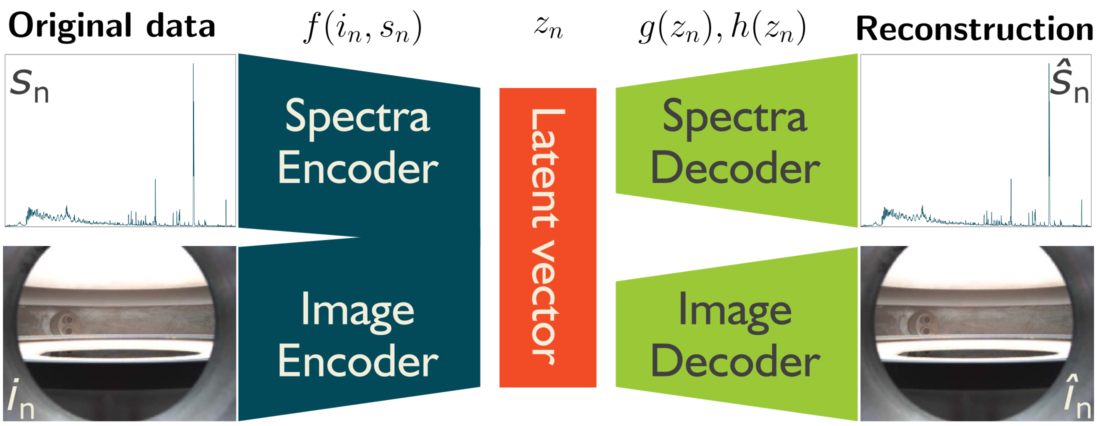


Figure 4. Autoencoder architecture block diagram.

Table 4. Settings for autoencoder model training and fine-tuning.

Config	Training	Finetune
Optimiser	Adam	Adam
Epochs	100	100
Base learning rate	2.5×10^{-4}	1×10^{-4}
Learning rate schedule	cosine decay	cosine decay
Warmup epochs	8	8
Warmup schedule	linear	linear
Batch size	2048	2048
Blocks	2,2,6,2	2,2,6,2
Filters (f)	64, 128, 256, 512	64, 128, 256, 512

The model is a multi-layer perceptron, a stack of identical dense neural network layers, trained with the latent representations, **z**, as a supervised objective. As we do not have a reference architecture for this model, and since its small size and low complexity mean it is fast to train, we used KerasTuner [58] to carry out a multi-objective Bayesian-optimisation of the number of dense layers, number of neurons and the learning rate for each of models with $l = [4, 16, 32, 64]$. We considered using the top five models as an ensemble, but we did not see a discernible improvement.

3.2. Evaluating the quality of unsupervised models

It is inherently difficult to evaluate the quality of unsupervised models as we do not have direct access to the objective that we are optimising for. In this work we trained our models to reduce the MSE between the original image and spectra and their reconstructions. However, this does not tell us if our latent space has useful information, i.e. if the encoding into this space is a useful empirical model of plasma information contained in the diagnostic data and/or if the latent representations produces by our tool model project back to the correct diagnostic information.

To evaluate this we have to create surrogate objectives that we believe provide us some insight into how well we achieve our underlying objective. The simplest method is to look at the performance of our models on our hold-out test data, if the model has simply memorised the input data and cannot generalise and interpolate between the trained data we will see poor reconstructions of the test data. To evaluate if our latent representation is useful for generating synthetic data we can look at the distribution of points in the latent space and make subjective judgements, e.g. large gaps and spaces between points are areas that cannot be sensibly interpolated across by our generative decoder. To evaluate the empirical quality of the models we can evaluate their behaviour around known mode transitions like the E–H mode, comparing trends to previous experimental data and changes in gas stoichiometry.

4. Properties of the latent space

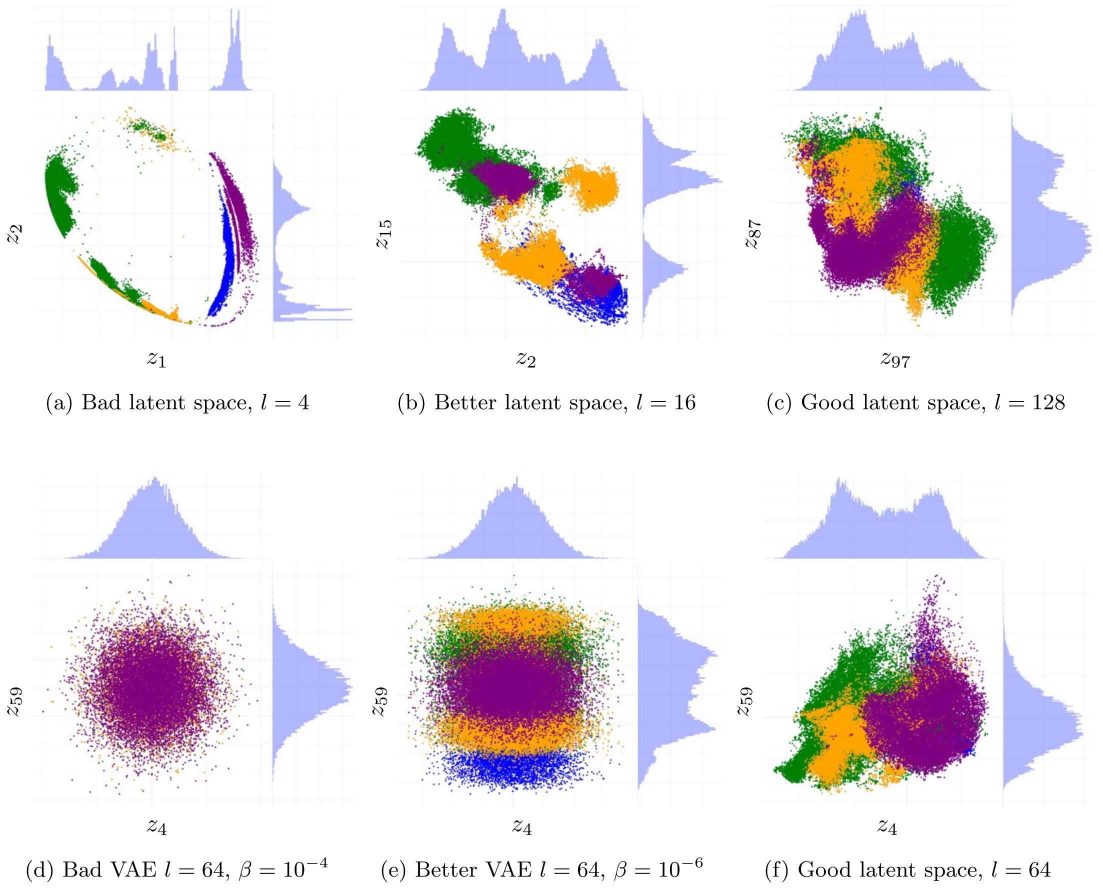
The overall aim of latent space modelling is to project input data onto a manifold in the latent space while preserving information and relationships within the data that are physically real and sensible, whilst not overfitting on spurious relationships that are not physically real or sensible. To make our latent representation usable we would like it to have some properties, for points to be close to a normal distribution, for points that are close in the real space (i.e. two plasmas that are similar to each other) to be close in the latent space and the reverse to be true, and for the latent space to be interpolatable, i.e. we can smoothly move through the latent space from one area to another without sharp discontinuities.

Many of these properties can be gained by simply using a large enough DL model with enough data. Large neural networks are inherently self-regularising [59] and with increasing size, reach a point where their outputs become Lipschitz continuous [60]. When training generative models on existing benchmark datasets, it is possible to use measures of image similarity to evaluate the performance of the model, such as the Fréchet inception distance [61]. However, these use pre-trained image classification networks to evaluate the quality of generated images. If our data was similar to the data used to train the classification network these methods can be used, or if you have some labelled data you can fine-tune one of these models for this use case. However, an OES of an Argon plasma has little similarity to images of planes and cats (which are typically employed in pre-trained networks) so we would not have any guarantee that these methods would work. This is an area of active research in the field of generative modelling and so in time new evaluation methods may appear that overcome this issue.

Without a quantitative measure of performance we are left with qualitative evaluations of our generative capabilities. The simplest is to look at the distribution of points in the latent space. If our model and dataset are large enough and the model is well trained, our latent space should be well behaved—close to a normal distribution and interpolatable. In figure 5 we show examples of the latent space of trained models, 'bad', 'better' and 'good'. The bad example shows a latent space that is extremely sparse and has significant spikes in the concentration of points, it would be very difficult to interpolate between points in this space as it has significant discontinuities and no meaningful representation moving off the central axis the points are stretched across. In the better example most of the points are reasonably close, although we have a strongly multimodal distribution and has separated into two clusters that would be extremely difficult to interpolate between. The good representation shows what we are looking for, our points are more smoothly distributed and there are no discontinuities within the latent space itself.

Unfortunately we cannot always expect our data to be perfectly well behaved like our 'good' representation. We cannot rely on the assumption that our data is independent and identically distributed. The conditions of one plasma are affected by the history of plasmas within that tool and we expect our latent space to encode some physically real multi-modal distributions, like E–H mode transitions, different gas stoichiometries and pressure regimes. Figure 5(f) shows a 'good' representation, the latent space is smooth and interpolatable, but one dimension has a bimodal distribution. We expect to see different physical modes in the data form independent normal distributions in the latent space and as long as it is physically possible to transition between these modes, and we have data covering the mode transition, the latent space can be used to interpolate between these modes.

In section 1.2 we discussed the VAE in comparison to the ordinary AE. In figures 5(d) and (e) we show VAE's trained with the same architecture and training settings as the $l = 64$ model, where β is the weighting factor between the reconstruction error and the KL divergence. The VAE training objective forces a the latent space to follow a normal distribution with a diagonal covariance matrix, $\mathbf{z} = \mathcal{N}(\mathbf{z}; 0, \mathbf{I})$, which figure 5(d) does. But this prior is not the true prior of the underlying data and so enforcing this degrades the model to



(a) Bad latent space, $l = 4$ (b) Better latent space, $l = 16$ (c) Good latent space, $l = 128$

(d) Bad VAE $l = 64$, $\beta = 10^{-4}$ (e) Better VAE $l = 64$, $\beta = 10^{-6}$ (f) Good latent space, $l = 64$

Figure 5. Example 2D slices of the latent space with well distributed and poorly distributed points, (blue—O_2, purple—Ar, green—SF_6 and orange—CF_4).

the point where the reconstruction error is a high as the $l = 4$ model while having 16 times more parameters in the latent space. When we relax this condition, by reducing β to 10^{-6}, the reconstruction error begins to improve to the level of the $l = 16$ model but, as can be seen in figure 5(e), it achieves this by changing the distribution of the latent space towards that of the $l = 64$ model.

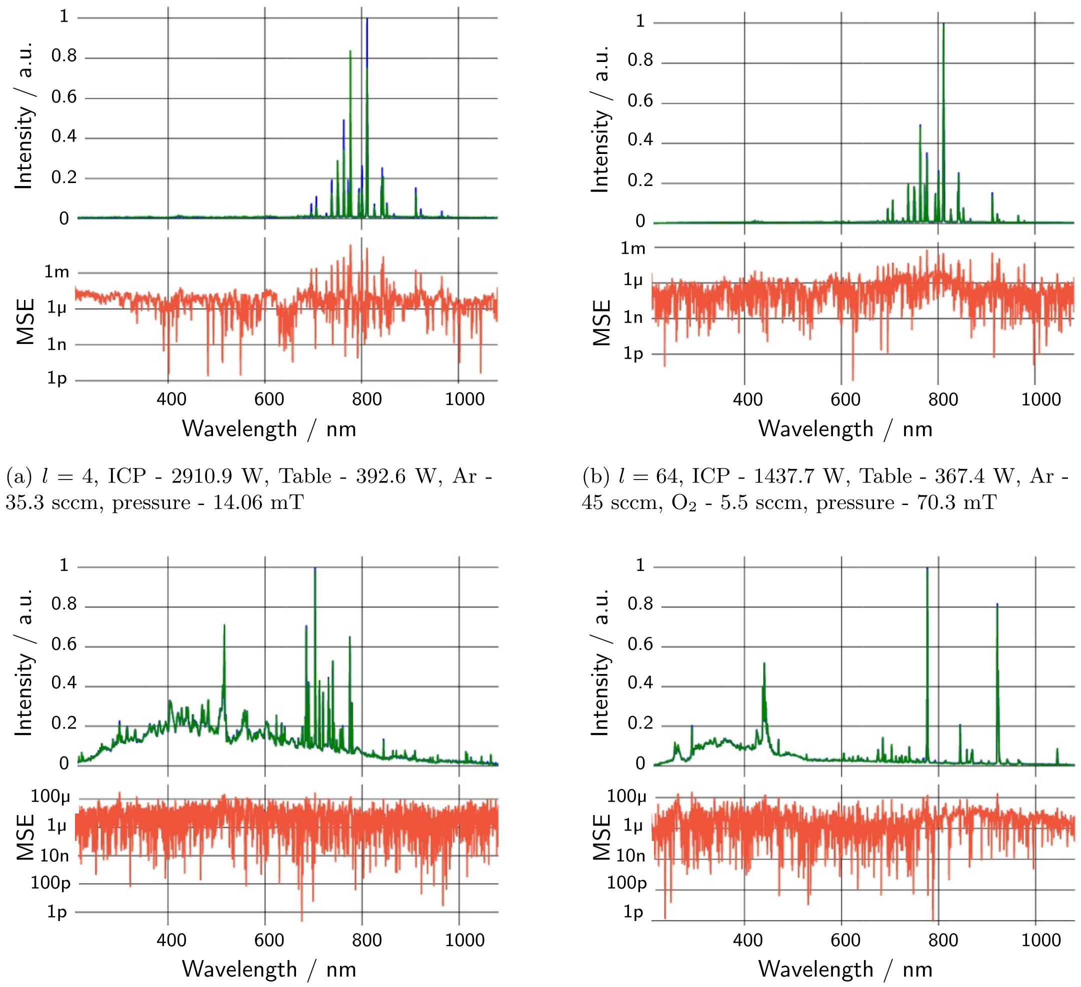
The requirement of the VAE prior for all dimensions of the latent space to be independent is also not entirely physically justifiable in real physical data. For example, if the underlying data contained information about electron density and density of atomic oxygen in the plasma these are not completely independent variables. An AE would be able to freely represent the functional relationships between these two variables. However, a VAE would require a latent dimension for each variable and each function that describes their relationship. The end result is a VAE that requires a far larger latent space to represent the relationships in the data and is far more likely to memorise the data as the latent space is not small enough to force the model to generalise.

5. Evaluating the generative model

A summary of the results from training the AE model is given in table 5. The training data split was used for directly training each model, the validation split was used to independently evaluate model performance for hyperparameter optimisation of the model learning rate. The optimal hyperparameters found for the training and fine-tuning step are summarised in table 4. The test split was kept as a holdout set for final model evaluation and was not used at any time during training and hyperparameter optimisation. The test and train errors are very close for all latent space sizes, indicating that the model has not overfit to the training data. In figures 6 and 7 we show 3 random examples, from the test split, $l = 64$, of the original and reconstructed data in each of our three gas mixtures and $l = 4$ for the Ar/O_2 example. The error on the reconstruction is extremely low for $l = 64$, but as can be seen in table 5 and figure 6(a), the reconstruction error decreases significantly for larger latent space size. In particular, figure 6(a) shows that the small latent space model makes significant errors in reconstructing the relative height of peaks in the spectrum and at $l = 64$ these are greatly minimised.

Table 5. Results of autoencoder model training.

	Spectra MSE			Image MSE		
Latent units (l)	Train	Validation	Test	Train	Validation	Test
4	2.09×10^{-4}	2.09×10^{-4}	2.06×10^{-4}	1.06×10^{-3}	1.10×10^{-3}	1.09×10^{-3}
8	4.50×10^{-5}	4.59×10^{-5}	4.71×10^{-5}	1.03×10^{-4}	1.04×10^{-4}	1.07×10^{-4}
16	2.83×10^{-5}	2.93×10^{-5}	2.87×10^{-5}	7.64×10^{-5}	7.56×10^{-5}	7.77×10^{-5}
32	1.29×10^{-5}	1.30×10^{-5}	1.31×10^{-5}	4.22×10^{-5}	4.19×10^{-5}	4.30×10^{-5}
64 (VAE $\beta = 10^{-2}$)	1.87×10^{-3}	1.91×10^{-3}	1.85×10^{-3}	2.21×10^{-2}	2.21×10^{-2}	2.15×10^{-2}
64 (VAE $\beta = 10^{-4}$)	1.62×10^{-4}	1.63×10^{-4}	1.62×10^{-4}	5.00×10^{-4}	4.96×10^{-4}	5.06×10^{-4}
64 (VAE $\beta = 10^{-6}$)	2.13×10^{-5}	2.17×10^{-5}	2.16×10^{-5}	7.33×10^{-5}	7.25×10^{-5}	7.43×10^{-5}
64	8.07×10^{-6}	8.25×10^{-6}	8.06×10^{-6}	3.69×10^{-5}	3.65×10^{-5}	3.74×10^{-5}
128	5.26×10^{-6}	5.37×10^{-6}	5.28×10^{-6}	3.27×10^{-5}	3.24×10^{-5}	3.12×10^{-5}



(a) l = 4, ICP - 2910.9 W, Table - 392.6 W, Ar - 35.3 sccm, pressure - 14.06 mT

(b) l = 64, ICP - 1437.7 W, Table - 367.4 W, Ar - 45 sccm, O_2 - 5.5 sccm, pressure - 70.3 mT

(c) l = 64, ICP - 1371.6 W, Table - 398.4 W, O_2 - 6.3 sccm, CF_4 - 62.8 sccm, pressure - 52.4 mT

(d) l = 64, ICP - 1924.8 W, Table - 369.5 W, O_2 - 20.1 sccm, SF_6 - 25.1 sccm, pressure - 8 mT

Figure 6. Measured OES and reconstructions in Ar/O_2 for $l = 4$ and 64, CF_4/O_2, and SF_6/O_2 plasmas, green line is the measured spectrum, blue line is the reconstructed spectrum. Given the difficulty of telling them apart, the red line below shows the mean squared error at each wavelength.

To evaluate the quality of our model's latent space we can look at the distribution of points encoded into the latent space. In figure 8 we can see the type of distributions we have in our latent space for $l = 8$ and 64. We can make a qualitative assessment of the quality of the latent space for generative modelling. For $l = 8$ the distributions show some sections that are smoothly and normally distributed, but has a large number of discontinuities (spikes and troughs) and are all strongly multimodal. For $l = 32$ and 64 some of our latent dimensions have a uni-modal distribution, but the majority have multi-modal distributions, and there is some complexity in the distributions. There are spikes present in $l = 32$ suggesting that some mode collapse

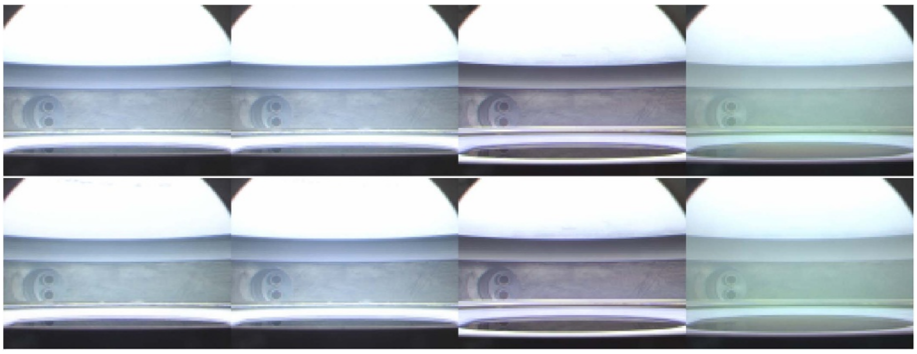
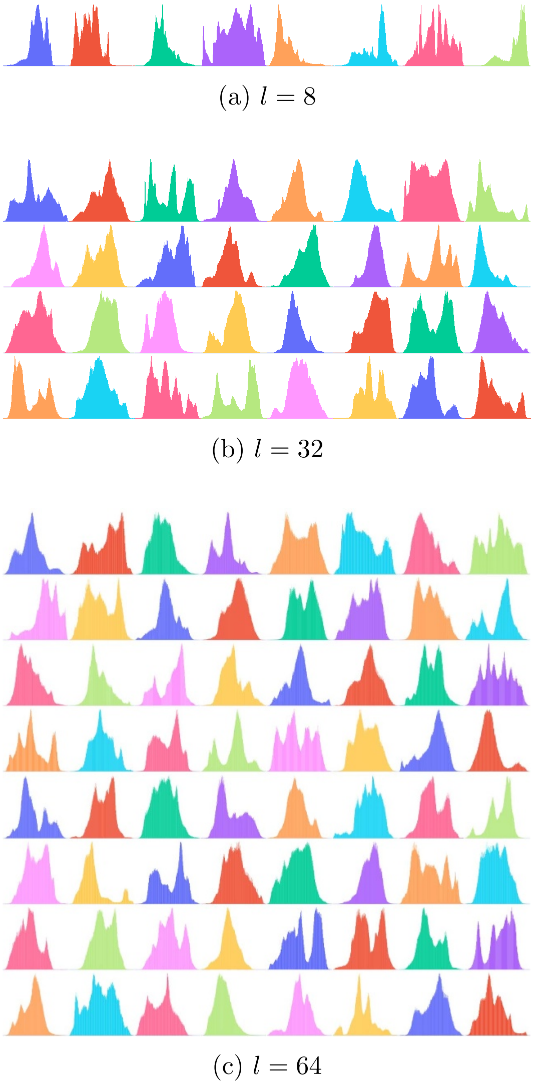


Figure 7. Measured images and reconstructions in Ar/O_2 ($l = 4$, $l = 64$), CF_4/O_2, and SF_6/O_2 plasmas. Top row is original images, bottom is reconstructions.



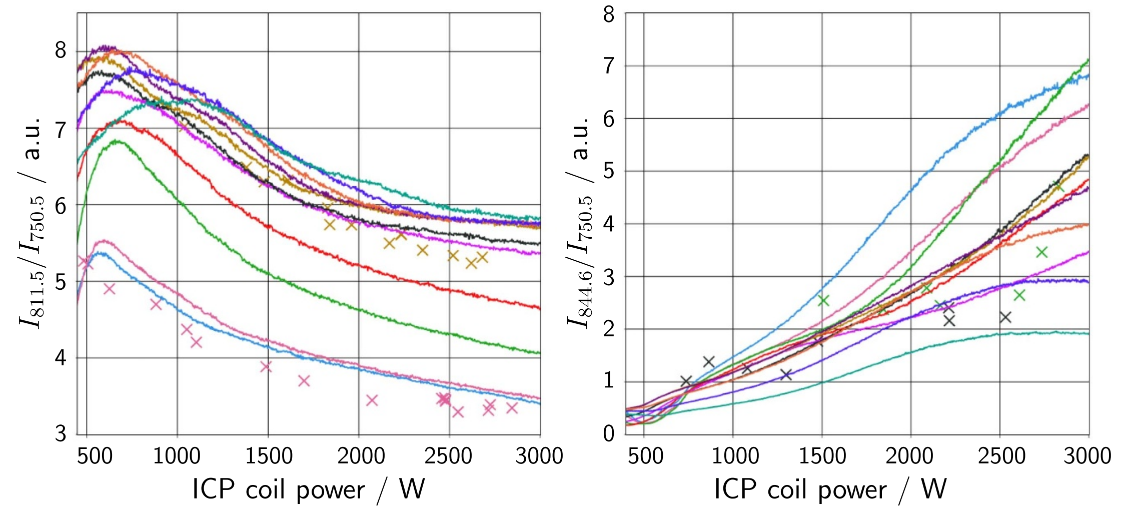
(a) $l = 8$

(b) $l = 32$

(c) $l = 64$

Figure 8. Histograms of the distribution of points in each latent dimension space for all image spectra pairs in the test set.

has occurred (e.g. multiple measurements mapped to the exact same place in the latent space), but not $l = 64$. In $l = 64$ there are no gaps in the latent space, although there are areas of very low density of points between parts of the distribution in a few of the latent dimensions, but $l = 32$ does have two areas of nearly zero density, suggesting a gap in the latent space. The $l = 128$ model has lower reconstruction error than the $l = 64$ model, but the latent spaces appear to have the same qualitative quality, without a quantitative assessment we will ere on the side of using the smallest latent space that appears qualitatively good. These qualitative



(a) Ar 811.5 nm / 750.4 nm line ratio (b) O_2 844.6 nm / Ar 750.4 nm line ratio

Figure 9. Sweep across from 400–3000 W ICP power in Argon and Ar/O_2, lines are generated data and crosses are measured points with 10% of the swept values. Sweeps repeated at 5 mT, 10 mT, 20 mT, 30 mT, 40 mT, 50 mT, 60 mT, 70 mT, 80 mT, 90 mT and 100 mT

assessments suggest that our $l = 64$ model can be used for generative modelling as we can smoothly interpolate between different areas of the latent without discontinuities, but the smaller $l = 8$ is unsuitable and $l = 32$ would be suitable for most areas, but would struggle around its discontinuities.

6. Results of synthetic experiments

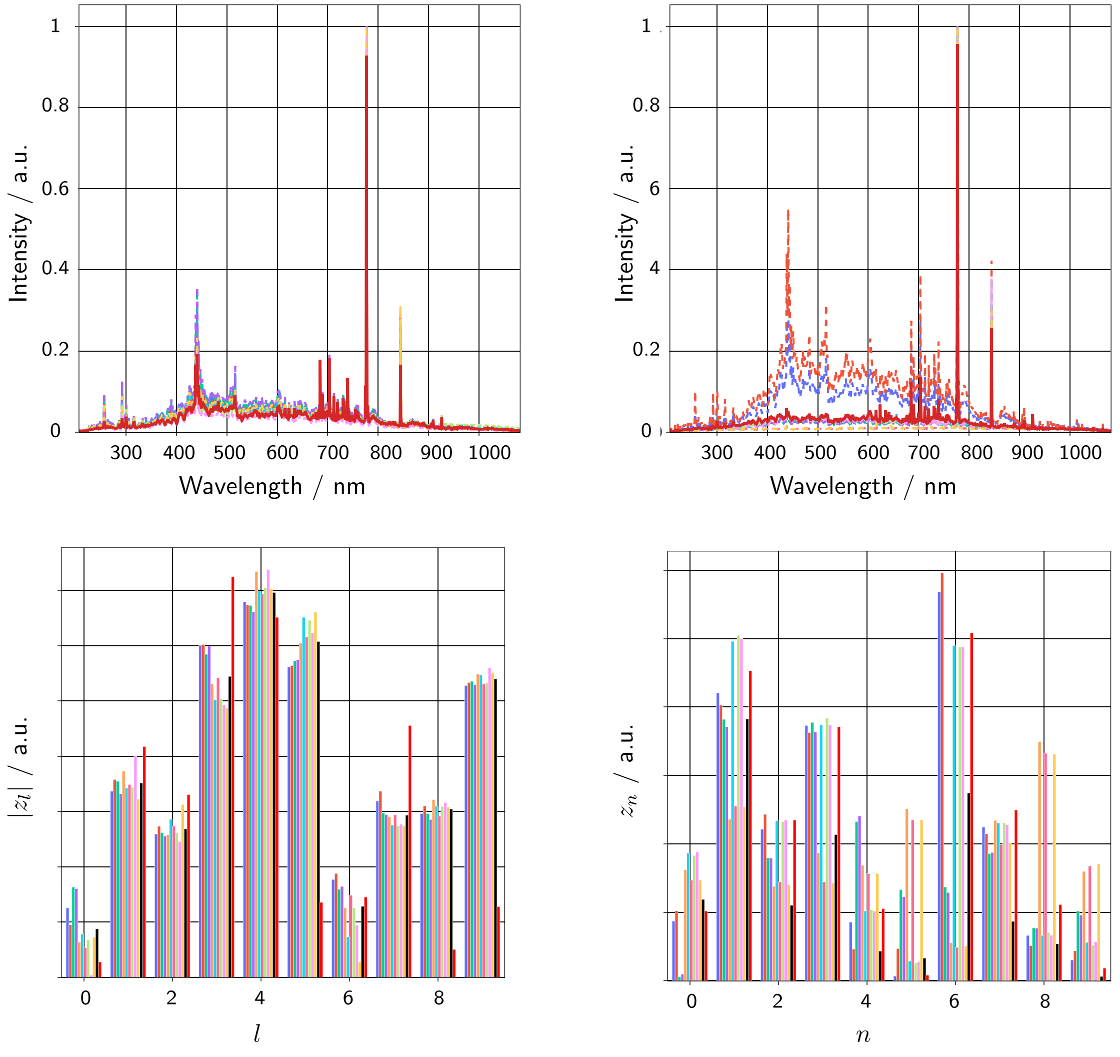
To carry out a synthetic experiment we use our tool-to-latent model, $\mathbf{z} = f(\mathbf{t})$, to produce latent representations, $\mathbf{z}$, and our two decoder branches, $\mathbf{i} = g(\mathbf{z})$, $\mathbf{s} = h(\mathbf{z})$, to generate spectra and images. We can generate an image spectra pair for one experiment point in 0.13 s/0.79 s on GPU/CPU, can compute a batch of 128 points in 0.25 s/51.22 s and a batch of 1024 in 1.34 s on an A100 GPU. In the simplest form, we can generate the expected spectra and image at a desired set of powers, pressures and gas mixture. We can also simply perform more complex experiments where we sweep across parameters in fine steps very quickly. Figure 9 shows a simple experiment where we sweep from 400–3000 W applied to the ICP source, 1024 steps, in pure argon and 8 sccm Ar, 50 sccm O_2 at different 11 pressures from 5–100 mT. We plot the line ratio of the Ar 811.5 nm and 750.4 nm lines in pure Ar and the ratio of the O_2 844.6 nm and Ar 750.4 nm lines in the Ar/O_2 mixture.

In figure 9(a) we show the variation in $(I_{811.5}/I_{750.5})$ ratio with power at pressures between 5 and 100 mT, at 10 and 60 mT we also plot the ratio at points in the data set that are close to the sweep. We can see that the points in the data are reasonably close to the generated data and follow the same trend. The overall trend in the data is in agreement with other experimental data by Czerwiec and Graves [62], although their reactor was a significantly different geometry. The trend in power shows a linear rise to the E–H mode transition point around 500–600 W and then decreases. Their data is at higher pressures, above 100 mT, and shows no change with pressure, our model shows a strong trend in an increase in $(I_{811.5}/I_{750.5})$ from 10–40 mT, then showing similar behaviour with little change with increasing pressure.

In figure 9(b) we show the variation in $(I_{844.6}/I_{750.5})$ ratio with power at pressures between 5 and 100 mT, at 20 and 50 mT we also plot the ratio at points in the data set that are close to the sweep. The points in the data show general agreement with the trends in the data, but the scatter in the points is quite high. The overall trend in the $(I_{844.6}/I_{750.5})$ ratio is in good agreement with earlier work by Fuller *et al* [63] with a relatively linear rise with applied power.

7. Limitations of the model and future work

The encoder model is able to embed any image / spectra pair into the latent space and very accurately decode them back into the real measurement space. Differences between the real plasma conditions of these measurements are represented by different coordinates in the latent space. When using the encoder model to monitor a plasma, the latent space representation will capture dynamic changes in the plasma over time.



(a) 20 mT

(b) 40 mT

Figure 10. Generated spectra and first ten latent coordinates at 2400 W ICP, 300 W table, 10 sccm O_2, 10 sccm CF_4 and spectra of ten nearest points in the dataset. In the upper plots, the solid line is the generated spectra and the dotted lines are the nearest 10. In the lower plots the first ten bars are the latent coordinates of the ten nearest, the black line is their average and the red is the generated latent.

However, our tool to latent model is very simplistic, it can only map a set of powers, gas flows and pressures to their average coordinate in the latent space, it cannot capture any dynamics.

We show an example of this in figure 10, at two pressures in a CF_4/O_2 plasma, we show the spectra generated at the latent coordinate produced by the tool to latent model and the ten nearest spectra to this point in the dataset. At 40 mT there is a high variation around the SiF emission peak at 443 nm as each point will have had a different history and the etch rate, in this reactor geometry, is more sensitive to input variations at higher pressures. This is reflected in the latent representations of these different plasmas, but our tool encoder finds a latent representation that produces an average of these spectra. At 20 mT, there is much less variation in the SiF emission peak and latent representation and so there is close agreement between all measurements and generated spectra.

The AE model itself does not have any sensitivity to variations over time as it only uses diagnostics that were gathered simultaneously. This limitation is by design. It allows the process of extracting information from the diagnostics to be separated from other tasks such as virtual metrology or predicting temporal variations in a plasma etcher over time. This represents one of the advantages of unsupervised learning, it allows us to easily separate different parts of a problem and combine the parts of our AE with different models to achieve different goals. These models can be trained with different data sources, where data much more limited or measurements more difficult without compromising the model performance by not having the quantity and diversity of data to train large deep models.

These tasks can use the latent representation as an input without needing enough labelled data to train a large model to understand the diagnostics. For example, in a time-series modelling task, the latent representation, of the image and spectra data, over time can be used as the model input and will act as a pre-trained feature extractor for the task. To improve the performance of this generative model we could replace our simple tool-to-latent model with a more complex model to account for trajectory of powers and pressures in the experiment. This could be achieved with a sequence-to-sequence model, with a sequence of tool parameters as the input and the corresponding sequence of latent representations as their labels.

8. Open source release of the dataset, trained models and code

The underlying dataset, configured as the train/validation/test splits used in the paper, is released under the Creative Commons Attribution 4.0 International [64]. The model code and trained models are available here and are released under the MIT license. An example notebook of using the model is available here and is released under the MIT license.

9. Conclusion

We have demonstrated that recent advances in generative modelling can be applied to optical diagnostics in low-temperature plasmas. These approaches require a heavily automated approach to experiments, to allow large amounts of data to be gathered in a reasonable amount of time. Large AE models can be trained, using existing open source libraries and model architectures, for a low cost on cloud GPUs or in a relatively short time on local GPU clusters.

We have shown that the latent space of AEs, trained on real plasma diagnostic data, is very sensitive to the size of the latent space. Any implicit bias to produce a model with the smallest number of parameters must be balanced by ensuring that the latent space is smooth and interpolatable if we want the model to be useful or have any capacity for generalisation.

Once trained, these AEs provide a low-cost method to generate large volumes of synthetic data for use in other work, such as validating or creating models. This is achieved by training an additional model to sample the latent space in the way required for the synthetic experiment. We have demonstrated this capability with a simple model to map tool inputs into the latent space and generate synthetic data that shows good agreement with experimental data in Argon and Ar/O_2 plasmas.

Large AEs can become a foundational building block for a wide array of plasma physics experiments and models when trained with large datasets of simple, but information dense diagnostics. The encoder can produce latent representations of diagnostics that are smoothly interpolatable and sensibly separates similar and dissimilar plasmas. These latent representations can be used for monitoring experiments or as inputs for other predictive models. The decoder can produce realistic and accurate data from latent representations and can be extended with auxiliary models to make a powerful generative model for synthetic experiments, which we aim to exploit in future work.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://doi.org/10.5281/zenodo.7704879.

Acknowledgments

This work was supported by the Engineering and Physical Research Council (Grant Number EP/L016389/1) and Oxford Instruments Plasma Technology (OIPT). The authors would like to thank Dr Thomas Miller and Mr Kris Balo at OIPT for their assistance in modifying the plasma processing tool for these experiments. We would also like to thank Dr Frazer Anderson at OIPT for giving us access to the labs and GPU cluster of OIPT's Innovation Development Group for this work.

ORCID iDs

G A Daly https://orcid.org/0000-0002-9349-8551
J E Fieldsend https://orcid.org/0000-0002-0683-2583
G R Tabor https://orcid.org/0000-0003-3549-228X

References

[1] Shi L, Valeo E J, Tobias B J, Kramer G J, Hausammann L, Tang W M and Chen M 2016 *Rev. Sci. Instrum.* **87** 11D303

[2] Jacobsen A S *et al* (the ASDEX Upgrade team) 2016 *Plasma Phys. Control. Fusion* **58** 045016

[3] Dalsania N, Patel Z, Purohit S and Chaudhury B 2021 *Fusion Eng. Des.* **171** 112578

[4] Juven A, Aumeunier M H, Brunet R, Bohec M L, Adel M, Miorelli R, Artusi X and Reboud C 2022 Temperature estimation in fusion devices using machine learning techniques on infrared specular synthetic data *2022 IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)* (*Nafplio, Greece*) pp 1–5

[5] Siminos E, Skupin S, Sävert A, Cole J M, Mangles S P D and Kaluza M C 2016 *Plasma Phys. Control. Fusion* **58** 065004

[6] Crilly A J, Appelbe B D, McGlinchey K, Walsh C A, Tong J K, Boxall A B and Chittenden J P 2018 *Phys. Plasmas* **25** 122703

[7] Milder A L, Ivancic S T, Palastro J P and Froula D H 2019 *Phys. Plasmas* **26** 022711

[8] Lewis W E, Knapp P F, Slutz S A, Schmit P F, Chandler G A, Gomez M R, Harvey-Thompson A J, Mangan M A, Ampleford D J and Beckwith K 2021 *Phys. Plasmas* **28** 092701

[9] Rodimkov Y, Bhadoria S, Volokitin V, Efimenko E, Polovinkin A, Blackburn T, Marklund M, Gonoskov A and Meyerov I 2021 *Sensors* **21** 6982

[10] Boffard J B, Jung R O, Lin C C and Wendt A E 2010 *Plasma Sources Sci. Technol.* **19** 065001

[11] Liu N, Zhong H, Lin Y, Chen T Y, Wang Z and Ju Y 2022 OH concentration and temperature measured by femtosecond cavity enhanced absorption spectroscopy in a nanosecond-pulsed dielectric barrier discharge *AIAA SCITECH 2022 Forum* (*San Diego, 3–7 January 2022*) (https://doi.org/10.2514/6.2022-1946)

[12] Gergs T, Borislavov B and Trieschmann J 2022 *J. Vac. Sci. Technol.* B **40** 012802

[13] Bond-Taylor S, Leach A, Long Y and Willcocks C G 2022 *IEEE Trans. Pattern Anal. Mach. Intell.* **44** 7327–47

[14] Frid-Adar M, Klang E, Amitai M, Goldberger J and Greenspan H 2018 Synthetic data augmentation using GAN for improved liver lesion classification *2018 IEEE 15th Int. Symp. on Biomedical Imaging (ISBI 2018)* (*Washington, DC, USA, 4 April 2018)* pp 289–93

[15] Cheng Y, Gong Y, Liu Y, Song B and Zou Q 2021 *Brief. Bioinform.* **22** bbab344

[16] Tempke R and Musho T 2022 *Commun. Chem.* **5** 1–10

[17] Lopez-Martin M, Carro B and Sanchez-Esguevillas A 2019 *Knowl. Inf. Syst.* **60** 569–90

[18] Choi K, Hawthorne C, Simon I, Dinculescu M and Engel J 2020 Encoding musical style with transformer autoencoders *Proc. 37th Int. Conf. on Machine Learning* (PMLR) pp 1899–908

[19] Ramesh A, Dhariwal P, Nichol A, Chu C and Chen M 2022 Hierarchical text-conditional image generation with CLIP latents (arXiv:2204.06125)

[20] Witman M, Gidon D, Graves D B, Smit B and Mesbah A 2019 *Plasma Sources Sci. Technol.* **28** 095019

[21] Mesbah A and Graves D B 2019 *J. Phys. D: Appl. Phys.* **52** 30LT02

[22] Ferreira D R, Carvalho P J and Fernandes H 2020 *IEEE Trans. Plasma Sci.* **48** 36–45

[23] Tello G, Al-Jarrah O Y, Yoo P D, Al-Hammadi Y, Muhaidat S and Lee U 2018 *IEEE Trans. Semicond. Manuf.* **31** 315–22

[24] Cheon S, Lee H, Kim C O and Lee S H 2019 *IEEE Trans. Semicond. Manuf.* **32** 163–70

[25] O'Leary J, Sawlani K and Mesbah A 2020 *IEEE Trans. Semicond. Manuf.* **33** 72–85

[26] Shojaei K and Mangolini L 2021 *J. Phys. D: Appl. Phys.* **54** 265202

[27] Boyer M D, Kaye S and Erickson K 2019 *Nucl. Fusion* **59** 056008

[28] Maggipinto M, Beghi A, McLoone S and Susto G A 2019 *J. Process Control* **84** 24–34

[29] Kingma D P and Welling M 2014 Auto-encoding variational bayes (arXiv:1312.6114 [cs, stat])

[30] Goodfellow I J, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y 2014 Generative adversarial networks (arXiv:1406.2661)

[31] Ho J, Jain A and Abbeel P 2020 Denoising diffusion probabilistic models *Advances in Neural Information Processing Systems* vol 33 pp 6840–51

[32] Dhariwal P, Jun H, Payne C, Kim J W, Radford A and Sutskever I 2020 Jukebox: a generative model for music (arXiv:2005.00341)

[33] Nitzan Y, Bermano A, Li Y and Cohen-Or D 2020 *ACM Trans. Graph.* **39** 225:1–14

[34] Yu J *et al* 2022 Scaling autoregressive models for content-rich text-to-image generation (arXiv:2206.10789)

[35] Rombach R, Blattmann A, Lorenz D, Esser P and Ommer B 2022 High-resolution image synthesis with latent diffusion models *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (*CVPR*) p 12

[36] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (MIT Press)

[37] van den Oord A, Vinyals O and Kavukcuoglu K 2017 Neural discrete representation learning *Advances in Neural Information Processing Systems* vol 30

[38] He J, Spokoyny D, Neubig G and Berg-Kirkpatrick T 2018 Lagging inference networks and posterior collapse in variational autoencoders *Int. Conf. on Learning Representations*

[39] Kingma D P, Salimans T, Jozefowicz R, Chen X, Sutskever I and Welling M 2016 Improved variational inference with inverse autoregressive flow *Advances in Neural Information Processing Systems* vol 29

[40] Tomczak J and Welling M 2018 VAE with a VampPrior *Proc. Twenty-First Int. Conf. on Artificial Intelligence and Statistics* (PMLR) pp 1214–23

[41] Razavi A, van den Oord A and Vinyals O 2019 Generating diverse high-fidelity images with VQ-VAE-2 *Advances in Neural Information Processing Systems* vol 32

[42] Ghosh P, Sajjadi M S M, Vergari A, Black M and Scholkopf B 2019 From variational to deterministic autoencoders *Int. Conf. on Learning Representations*

[43] Ghose A, Rashwan A and Poupart P 2020 Batch norm with entropic regularization turns deterministic autoencoders into generative models *Proc. 36th Conf. on Uncertainty in Artificial Intelligence* (*UAI*) (PMLR) pp 1079–88

[44] Daly G A, Fieldsend J E and Tabor G 2022 Variational autoencoders without the variation (arXiv:2203.00645)

[45] Maggipinto M, Masiero C, Beghi A and Susto G A 2018 *Proc. Manuf.* **17** 126–33

[46] Zhang H, Wang P, Gao X, Gao H and Qi Y 2020 Automated fault detection using convolutional auto encoder and k nearest neighbor rule for semiconductor manufacturing processes *3rd Int. Conf. on Intelligent Autonomous Systems (ICoIAS)* (*Singapore, 26–29 February 2020)* pp 83–87

[47] Jaeckel P 2002 *Monte Carlo Methods in Finance* (Wiley)

[48] Morokoff W J and Caflisch R E 1994 *SIAM J. Sci. Comput.* **15** 1251–79

[49] Sobol' I M 1967 *USSR Comput. Math. Math. Phys.* **7** 86–112

[50] Blackman R B and Tukey J W 1958 *Bell Syst. Tech. J.* **37** 185–282

[51] LeCun Y, Bottou L, Orr G B and Müller K R 1998 Efficient BackProp *Neural Networks: Tricks of the Trade* (*Lecture Notes in Computer Science*) ed G B Orr and K R Müller (Springer) pp 9–50

[52] Ioffe S and Szegedy C 2015 Batch normalization: accelerating deep network training by reducing internal covariate shift *Proc. 32nd Int. Conf. on Machine Learning* (*Proc. Machine Learning Research*) vol 37 pp 448–56

[53] Liu Z, Mao H, Wu C Y, Feichtenhofer C, Darrell T and Xie S 2022 *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* pp 11976–86

[54] Kingma D P and Ba J 2015 Adam: a method for stochastic optimization *3rd Int. Conf. on Learning Representations, ICLR 2015* (*San Diego, CA, USA, 7–9 May 2015, Conf. Track Proc.*) ed Y Bengio and Y LeCun

[55] Loshchilov I and Hutter F 2017 SGDR: stochastic gradient descent with warm restarts *5th Int. Conf. on Learning Representations, ICLR 2017* (*Toulon, France, 24–26 April 2017*) (Conference Track Proceedings (OpenReview.net))

[56] Chollet F *et al* 2015 Keras (available at: https://github.com/keras-team/keras)

[57] Abadi M *et al* 2015 TensorFlow: large-scale machine learning on heterogeneous systems software available from tensorflow.org (available at: www.tensorflow.org/)

[58] O'Malley T, Bursztein E, Long J, Chollet F, Jin H and Invernizzi L 2019 KerasTuner (available at: https://github.com/keras-team/keras-tuner)

[59] Barrett D and Dherin B 2020 Implicit gradient regularization *Int. Conf. on Learning Representations*

[60] Bubeck S and Sellke M 2021 A universal law of robustness via isoperimetry *Advances in Neural Information Processing Systems* vol 34, ed M Ranzato, A Beygelzimer, Y Dauphin, P Liang and J W Vaughan pp 28811–22

[61] Heusel M, Ramsauer H, Unterthiner T, Nessler B and Hochreiter S 2017 GANs trained by a two time-scale update rule converge to a local nash equilibrium *Advances in Neural Information Processing Systems* vol 30

[62] Czerwiec T and Graves D B 2004 *J. Phys. D: Appl. Phys.* **37** 2827

[63] Fuller N C M, Malyshev M V, Donnelly V M and Herman I P 2000 *Plasma Sources Sci. Technol.* **9** 116

[64] Daly G, Fieldsend J, Hassall G and Tabor G 2023 Data-driven plasma modelling: Fluorocarbon ICP data set *Zenodo* (https://doi.org/10.5281/zenodo.7704879)