



Assignment of master's thesis

Title:	Machine Learning Techniques for Laser-Plasma Acceleration Optimization
Student:	Bc. Matěj Jech
Supervisor:	Mgr. Alexander Kovalenko, Ph.D.
Study program:	Informatics
Branch / specialization:	Knowledge Engineering
Department:	Department of Applied Mathematics
Validity:	until the end of summer semester 2024/2025

Instructions

Particle accelerators are powerful scientific instruments to study the fundamental properties of Matter and the Universe. One of the most significant discoveries made using particle accelerators is the Higgs boson, a fundamental particle that gives other particles mass. The Large Hadron Collider at CERN, the world's largest and most powerful particle accelerator, discovered the Higgs boson in 2012, confirming a key prediction of the Standard Model of particle physics.

On the other hand, the development of laser-plasma accelerators has enabled significant miniaturization of particle accelerators, making them indispensable in various scientific fields. These devices are workhorses in myriad fields of science, blasting out fundamental particles and generating intense beams of X-rays for studies of biomolecules and materials. Using laser-plasma accelerators, kilometers-long devices can be squished into a few centimeters and, for example, the Large Hadron Collider, the biggest and most powerful of them all, could fit in a classroom.

However, the laser-plasma acceleration process is highly nonlinear, and a lot of trial-and-error optimization is usually necessary to maximize the particle beam's power output. The machine operator relies on input data such as electromagnetic field intensity, backing gas pressure, plasma gradient of the shock wave, plasma filament expansion, and data obtained from cameras inside the machine to maximize the particle beam's output.



Therefore, using machine learning techniques could be promising in understanding the input-output relationship in the laser plasma acceleration process, potentially improving the performance of these new-generation accelerators. The work will be conducted in collaboration with a world-class team from International Laser Research Centre (<https://www.eli-beams.eu/>), where the 4 most powerful lasers in the world are located.

To successfully complete the diploma thesis the following research steps should be taken:

- Carry on a comprehensive literature review on the fundamentals of laser-plasma accelerators, machine learning techniques, and their applications in particle accelerators;
- Preprocess the data obtained from the scientific partners;
- Train a conditional generative model to broaden the scope of investigation for laser-plasma experiments.
- Explore the possibility of application of physics Informed machine learning in order to overcome possible challenges by incorporating physical laws and constraints into the machine learning algorithm;
- Model testing "in the wild";

Literature:

S. Jalas, et al. Bayesian Optimization of a Laser-Plasma Accelerator, <https://doi.org/10.1103/PhysRevLett.126.104801>

A. Döpp, et al. Data-driven Science and Machine Learning Methods in Laser-Plasma Physics, <https://doi.org/10.48550/arXiv.2212.00026>

R. J. Shalloo, et al. Automation and control of laser wakefield accelerators using Bayesian optimization, <https://doi.org/10.1038/s41467-020-20245-6>

Master's thesis

**MACHINE LEARNING
TECHNIQUES FOR
LASER-PLASMA
ACCELERATION
OPTIMIZATION**

Bc. Matěj Jech

Faculty of Information Technology
Department of Applied Mathematics
Supervisor: Mgr. Alexander Kovalenko, Ph.D.
January 11, 2024

Czech Technical University in Prague

Faculty of Information Technology

© 2024 Bc. Matěj Jech. All rights reserved.

This thesis is school work as defined by Copyright Act of the Czech Republic. It has been submitted at Czech Technical University in Prague, Faculty of Information Technology. The thesis is protected by the Copyright Act and its usage without author's permission is prohibited (with exceptions defined by the Copyright Act).

Citation of this thesis: Jech Matěj. *Machine Learning Techniques for Laser-Plasma Acceleration Optimization*. Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2024.

Contents

Acknowledgments	vi
Declaration	vii
Abstract	viii
Acronyms	ix
Introduction	1
1 Modeling of physical phenomena	3
1.1 Alpha campaign at ELI Beamlines	3
1.2 Usage of neural networks	4
2 Search space augmentation with generative AI	6
2.1 Characterizing the original search space	6
2.2 Mapping the search space	6
2.3 Generative AI	7
2.3.1 Evaluation	7
2.3.2 Applications	7
2.4 Transfer learning	8
3 Diffusion probabilistic models	9
3.1 Denoising Diffusion Probabilistic Model (DDPM)	9
3.2 Denoising Diffusion Implicit Model (DDIM)	10
3.3 Diffusion-based Autoencoders (DiffAE)	11
3.4 Classifier-free guidance	11
3.5 Inverse heat dissipation	12
4 Physics informed machine learning	14
4.1 Implementation of physics into the training process	14
4.1.1 Observational bias	14
4.1.2 Inductive bias	14
4.1.3 Learning bias	15
4.1.4 Hybrid approaches	15
4.1.5 Implementation for electron acceleration	15
4.2 Calculation of electron beam spectra	15
4.3 Penalisation based on electron beam position	16
5 Optimization	17
5.1 Bayesian optimization	17
5.2 Reinforcement learning	18

6	Architecture and implementation	20
6.1	Data preparation pipeline	20
6.2	Diffusion probabilistic model	22
6.2.1	Training loop	23
6.2.2	Sampling algorithm	23
6.2.3	Inverse heat dissipation	26
6.2.4	Transfer learning	26
6.3	Physics-informed loss penalization	27
6.3.1	Electron beam spectra simulation	27
6.3.2	Electron beam pointing pixel	27
6.4	Diffusion autoencoder	28
6.4.1	Architecture	28
6.4.2	Training loop	29
6.4.3	Sampling	29
7	Experiments and Results	30
7.1	Evaluation methodology	30
7.2	DDIM performance	31
7.2.1	Physics-informed performance	31
7.2.2	Analysis of noise steps	31
7.2.3	CFG statistics	33
7.2.4	Section count analysis	33
7.3	Sampling speed	36
7.4	Inverse heat dissipation (IHD)	39
7.5	DiffAE results	40
7.6	Evaluation of best configurations	40
7.6.1	Sampling of random data points	43
8	Future work	46
8.1	Modern diffusion methods	46
8.2	Autoencoder architecture	46
8.3	Expanding physics research	46
	Conclusion	48
	A Examples of model sampling performance	50
	Contents of enclosed medium	66

List of Figures

1.1	Experimental setup of the alpha campaign. (a) The L1-Allegra laser system. (b, c, e) Various laser-related diagnostics. (d) View of the focal spot of the laser. (f) The gas-jet target of the laser. (g) Electron spectrometer. (h) Electron-beam trace projected on the Lanex screen. [13]	4
3.1	Overview of the architecture of the DiffAE model from [44]	12
3.2	A comparison between Gaussian noising and blurring as introduced in [47].	13
6.1	Preprocessing pipeline.	22
6.2	Schematics of the building blocks of the diffusion model’s final architecture. Arrows suggest the input/output of tensors. Visualized in green are normalization layers, red are activation layers, blue are layers changing the scale of tensors and white are layers with no learnable parameters. Other layers with learnable parameters are in yellow.	24
6.3	Schematic of the full model architecture. Arrows suggest the input/output of data; red arrows symbolize skip connections. Tensor x contains image data and tensors t and y contain the noise step and settings respectively. Individual blocks are described in Figure 6.2.	25
6.4	Effect of the noising process. Images from left to right are in 100 noise steps increments from 0 to 700.	25
6.5	The effect of the inverse heat dissipation blurring. The input image in the top left is processed by 100 noise steps in each of the following figures going from left to right, top to bottom.	26
6.6	This figure presents an image from the training dataset, accompanied by its corresponding spectrum calculation. The image is rendered in monochrome, with a colormap applied to enhance visual clarity and facilitate easier interpretation.	28
7.1	Comparison of different model versions.	32
7.2	Influence of different noise steps values on metrics.	34
7.3	Influence of different CFG values on metrics.	35
7.4	Influence of different sections values on metrics.	37
7.5	Influence of noise steps and sampling batch size on the generation speed using the DDPM algorithm.	38
7.6	Impact of sections and sampling batch size on generation speed using the DDIM algorithm.	38
7.7	Showcase of performance of the IHD model. The left column shows images from the training dataset while the right shows generations given the same settings vector.	39
7.8	An input image with its electron beam spectrum (top row) and its 3 different stochastic variations.	41

7.9	Comparison of the performance of the two best models. The left column shows an average training image per validation experiment. “Model 1” is the <code>phys_spec_beam</code> model expected to perform best across all metrics and “Model 2” is the best model when variance difference weight is doubled. The intensity of the images is doubled for ease of viewing.	42
7.10	Comparison of sampling variance of the two best models. The left column shows random images from the validation folder. “Model 1” stands for the overall best <code>phys_spec_beam</code> model setting and “Model 2” is the one more focused on variance difference.	44
7.11	Comparison of model performance when changing the acquisition time parameter. The different values can be read on the left and models on the top. Model 1 is the generally best model, Model 2 is the variance-focused one. Each image is an average of 16 generated samples.	45

List of Tables

7.1	Best models according to the selected metrics. The numbers in brackets indicate the rank of the model in the corresponding metric.	40
-----	--	----

I would like to thank my friends, family and girlfriend for supporting me throughout my studies. Thank you to my supervisor Alex Kovalenko for his great advice and guidance throughout the implementation and writing of the thesis. A big thank you to the team at ELI-Beamlines - Gabriele, Carlo, Leo, Illia, Jirka, Bětka, Sebastian, Filip and Michal for creating a great working environment and for answering all of my physics related questions.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Czech Technical University in Prague has the right to conclude a licence agreement on the utilization of this thesis as a school work pursuant of Section 60 (1) of the Act.

In Prague on January 11, 2024

Abstract

The thesis deals with the analysis of data from the laser-plasma particle accelerator in collaboration with the scientific institution ELI Beamlines. In the scope of the work, a data pre-processing process was designed and a generative model simulating the course of physics experiments was developed. The model is conditioned on a vector of experimental parameters and generates image data showing the energy spectrum of the accelerated electron beam. The developed model can be used as a partial substitute for real experiments, which are costly in terms of time and finances. It can also be used as a simulation of real experiments for various optimization methods. This thesis defines the process of training and testing candidate models with three different architectures and based on four hyperparameters. The resulting model can generate data at a rate of 1.8 images per second and has been evaluated based on a number of metrics, including the expert opinion of scientists, as a trustworthy tool to simulate the electron acceleration process.

Keywords LWFA, generative AI, diffusion, physics informed ML, physics modelling

Abstrakt

Práce se zabývá analýzou dat z laser-plasmového urychlovače částic ve spolupráci s vědeckou institucí ELI Beamlines. V rámci práce byl navržen proces předpřípravy dat a vyvinut generativní model simulující průběh fyzikálních experimentů. Model je podmíněn vektorem parametrů experimentu a generuje obrazová data zobrazující energetické spektrum paprsku urychlených elektronů. Vyvinutý model lze využít jako částečnou náhradu skutečných experimentů, které jsou časově i finančně nákladné. Rovněž jej lze použít jako simulaci skutečných experimentů pro různé optimalizační metody. Práce definuje proces trénování i testování kandidátních modelů se třemi různými architekturami a na základě čtyř hyperparametrů. Výsledný model dokáže generovat data rychlostí 1.8 obrázků za sekundu a byl vyhodnocen na základě řady metrik včetně expertního názoru vědců jako věrohodný způsob simulace průběhu accelerace elektronů.

Klíčová slova LWFA, generativní AI, diffusion, fyzikálně informované strojové učení, modelování fyziky

Acronyms

AI Artificial Intelligence. 6, 7, 11, 20

CFG Classifier-free Guidance. 31, 33, 40

DDIM Denoising Diffusion Implicit Model. 10, 11, 23, 28, 29, 31, 33, 36, 40, 48

DDPM Denoising Diffusion Probabilistic Model. 9, 10, 20, 23, 36, 48

DiffAE Diffusion Autoencoder. 11, 28, 48

FID Fréchet Inception Distance. 7, 30, 31, 33, 36, 40, 48

IHD Inverse Heat Dissipation. 26, 39

ML Machine Learning. 8, 14

MSE Mean Squared Error. 10, 30, 31, 33, 36, 40, 48

PIC Particle-in-cell simulation. 3, 18

RL Reinforcement Learning. 18, 19

Introduction

This thesis combines two highly contested fields of machine learning and physics - image generation and particle acceleration. These two seemingly unrelated tasks can be joined to assist in the development of capable devices with large-scale applications in medicine and other advanced scientific research.

Process simulation by generative AI

Throughout recent years diffusion models [1] have emerged as a promising tool in image generation [2]. Their use extends well into the general public with a large amount of images produced daily by users using prompts as conditions for the generation. Many commercially produced models emerged [3, 4] along with some open-source solutions [5] offering users to train their own models or to finetune existing ones. The use case of these models is mostly for personal use or sometimes as a faster and cheaper alternative for creating advertisements or for other marketing practices. [6]

The main focus is however often narrow-mindedly centered only on the produced images viewed as singular products. Little attention is paid to the representation of the training data learned by the model. This is mostly because one of the main requirements for these models is for them to be as adaptable as possible, allowing the generation of any image imaginable. The process of finetuning [7] breaks this paradigm only partially as the resulting models are still capable of generating a large scope of images well, they are only specialized and therefore perform better on some specific subgenre or subdataset.

The big strength of generative models lies in their ability to learn the representation of data very precisely which is what allows the sampling process to produce such realistic outputs. If the model is trained on a specific dataset it is highly specialized in a certain field and can be used as a simulator for some real-world process that outputs image (or other) data [8, 9]. The model can also be conditioned based on some parameters specific to the simulated task to further increase its potential [10]. Such models can no longer be used as general image generators but serve as powerful tools in understanding complex real-world processes through the investigation of the learned representation of data.

Electron acceleration

Particle acceleration is a discipline in physics characterized by uncovering the basic nature of all objects through a detailed analysis of atoms' behavior at large velocities and when colliding with other particles. The acceleration is traditionally achieved in large devices spanning kilometers in circumference [11]. This limits their use to specialized facilities which limits their accessibility.

Laser wakefield acceleration (LWFA) [12] is a method of accelerating electrons by using complex interactions of laser and plasma. The main advantage to traditional accelerators is their compact size - capable to fit into a laboratory space spanning a few meters. These electron accelerators are characterized by a plethora of parameters along with a high sensitivity to their changes [13]. This makes the development and optimization of them a difficult task [14].

Simulation and optimization

The task presented in this thesis is to create a conditional generative model capable of simulating the complex non-linear processes of the laser-plasma electron accelerator. This model, conditioned by real-world device parameters, would be capable of estimating the spectra of accelerated electrons. These spectra in addition to the electron energy may contain a plethora of important information that can be used for device optimization, therefore traditional predictive modeling is not appropriate in this case. This model will then be used to optimize the real device without the costs of running real experiments and with superior time efficiency.

Modeling of physical phenomena

Physical phenomena are typically explained by an underlying theory. This theory often supposes ideal circumstances to make its definition clear and including every real-world aspect could be unfeasible. In experimental physics, these approximations can come to light and therefore the results expected from the theory could be more or less skewed or perhaps different entirely. Modeling of physical phenomena can be done for example by running Particle-in-cell simulation (PIC) simulations [15]. These can be very precise in following the underlying theory but can get very time and hardware-demanding to the point where running some simulations is not possible on the world's best computers.

1.1 Alpha campaign at ELI Beamlines

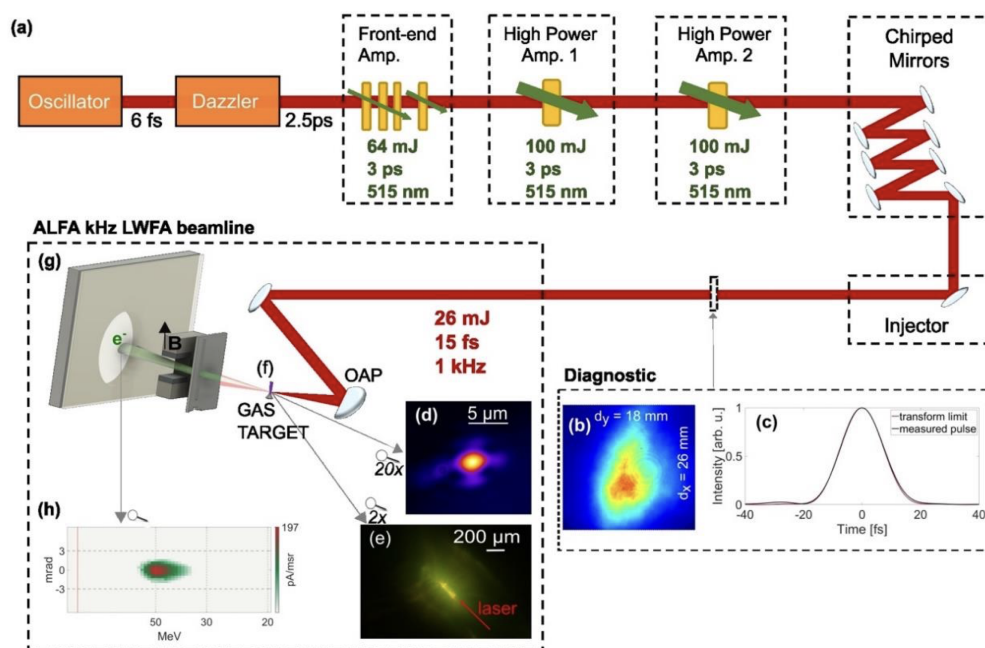
The experimental work central to this thesis was conducted at the ELI Beamlines facility, located in Dolní Břežany [16]. This facility is renowned for housing the world's most powerful lasers, which were instrumental in collecting the data utilized in this research.

The key apparatus in these experiments was the L1-Allegro [17] laser system, notable for its high-frequency operation at 1 kHz combined with a high pulse energy of around 26 mJ. The laser was directed toward a gas target, composed either of pure Nitrogen or a Nitrogen-Helium mixture. Upon laser interaction, the gas undergoes ionization, forming plasma. As the laser traverses this plasma, it generates a wakefield in its trail. Electrons within this plasma are then drawn into this wakefield and accelerated to high energies, essentially “surfing” [18] in a trajectory parallel to the laser's path. The accelerated electron beam is directed toward a spectrometer equipped with a Lanex screen [19]. The interaction of electrons with the screen results in luminescence at the points of impact, enabling their detection and capture via a camera [19].

A pivotal aspect of this setup is the incorporation of a strong magnet positioned before the screen. The magnet's role is to deflect the electron beam, causing a displacement relative to the beam's original path. By analyzing this displacement and applying a deflection curve specific to the experimental setup, it is possible to measure the energy levels of the electrons striking the screen [13]. This is how the image dataset used in this thesis was created. This technique is known as laser wakefield acceleration (LWFA) and is significant for its practical applications.

Electron accelerators are crucial in healthcare, notably in medical imaging [20, 21], radiation medicine [22, 23] and cancer treatment [24]. Additionally, they are invaluable in future research domains that would require accelerated particles, offering more compact and efficient alternatives to traditional accelerators.

During the course of these experiments, the scientists at ELI Beamlines successfully generated electron beams with energies reaching up to 50 MeV, additionally characterized by a low energy



■ **Figure 1.1** Experimental setup of the alpha campaign. (a) The L1-Allegro laser system. (b, c, e) Various laser-related diagnostics. (d) View of the focal spot of the laser. (f) The gas-jet target of the laser. (g) Electron spectrometer. (h) Electron-beam trace projected on the Lanex screen. [13]

spread [13]. A detailed illustration of the experimental setup employed at ELI Beamlines is depicted in Figure 1.1, providing a comprehensive visual overview of the entire process and apparatus.

1.2 Usage of neural networks

The physical theory explaining the conducted experiments is developed based on the results of the accelerator experiments. Due to the complex nature of the experiments, it does not provide a complete understanding of the underlying principles. The data collected in the Alpha experiments consists of three parts. A static part that doesn't change when input parameters remain the same. A dynamic part that explains all time-dependent parts such as the heating of the equipment, tidal forces, slow fatigue of material etc. The last part is the stochastic element. This incorporates the truly random influences that cannot be explained by anything deliberate or predictable.

The experiments in laser-plasma interaction are influenced by a multitude of input parameters, encompassing both laser settings and the nuances of plasma creation. These parameters are not only numerous but also interdependent, creating a complex web of influences. This intricate interplay, coupled with the substantial impact of even minor parameter adjustments, results in a highly non-linear system. The resultant characteristics of the electron beam are therefore a product of this complex, multifaceted interaction, underscoring the challenges in predicting and controlling the experimental outcomes. [13, 14]

Neural networks, with their ability to approximate any well-behaved function at certain conditions [25], are particularly well-suited for modeling physical phenomena. They can learn the static part of the data that follows theoretical predictions, the dynamic or temporal part that changes over time, and the stochastic part that represents random variations.

Neural networks consist of layers of learnable parameters. By adjusting these parameters during training, they can learn to represent complex, non-linear relationships between inputs and

outputs. This makes them powerful tools for modeling physical phenomena, where the underlying processes can often be highly non-linear.

Moreover, neural networks are robust to noise in the data. During training, they learn to recognize and ignore random variations that do not contribute to the underlying pattern. This attribute is called feature or representation learning [26, p. 524] and allows the network to extract meaningful parts of the data called features. This is particularly useful when dealing with experimental data, which can often be noisy.

In addition, neural networks can handle temporal dynamics. Recurrent neural networks (RNNs) [27] and their variants, such as Long Short-Term Memory (LSTM) [28] networks, are designed to process sequential data and can capture temporal patterns and dependencies. This makes them suitable for modeling dynamic aspects of physical phenomena.

Finally, neural networks are highly adaptable. As new data becomes available, the network can be retrained to incorporate this information. This allows the model to evolve and improve over time, which is crucial in fields like physics where new experimental data is constantly being generated.

In [29] Streeter et al. explore a goal similar to this thesis. The authors develop an ensemble [26, p. 256] of Variational Autoencoders [30] to generate electron spectra. The authors use a recalculated one-dimensional representation of the spectra to reduce computing power. The conditional parameters include laser energy and several laser-related diagnostics. Each model of the ensemble would predict a spectrum based on the conditions and a resulting generation would be created by averaging all intermediate results. The deviation of these intermediates was used to assess the uncertainty of the prediction.

Search space augmentation with generative AI

The optimization of an experiment necessitates a comprehensive understanding of the underlying dynamics and the influence of each parameter. However, achieving this level of understanding is often infeasible without resorting to approximations. As discussed in the previous chapter, neural networks provide a robust framework for such approximations. Using a network to optimize a task with a multidimensional and highly non-linear search space requires a lot of data [31]. If the provided training dataset is not extensive, the search space is sparsely populated by datapoints. To increase the size of the training dataset, new samples can be taken from the real-world process that produces them. This option can be unfeasible due to the high monetary or time costs of data acquisition. This chapter focuses on overcoming this obstacle through augmenting and populating the search space with generative Artificial Intelligence (AI).

2.1 Characterizing the original search space

The optimization process begins with the approximation of the experiment using a function. This function, termed the objective function [26, p. 82], maps the parameters of the experiment to a metric that quantifies the success of the experiment. This objective function is what defines the search space. The exploration of this function's landscape is essential for identifying the optimal set of input parameters. The dimensionality and complexity of a search space are among its most defining characteristics. Complicated problems typically exhibit high-dimensional and highly non-linear search spaces, which pose significant challenges in terms of mapping and navigation [32]. The search for an optimal configuration involves locating an extremum (either a minimum or a maximum) within the problem's search space. High dimensionality implies a multitude of potential directions to explore from each point in the space. Non-linearity suggests that even small changes in one input parameter require redefining the choice in other parameters through their intricate cooperative influence on the objective function.

2.2 Mapping the search space

The acquisition of an exact landscape of the search space is typically unattainable. This is particularly true for accelerator optimization due to its high dimensionality and non-linearity [14]. The construction of the search space and the identification of an optima require a substantial number of data points. The requirement for these points escalates with increasing non-linearity

and dimensionality [31]. Each of these points corresponds to a real experiment conducted with specific parameter values, with the results being logged. However, access to the laser is limited to a few weeks throughout the year, some of which are dedicated solely to setup and alignment. This leaves a narrow window for conducting experiments for data collection.

Owing to this limitation, the search space is mapped by a sparse set of real-world data. This is insufficient for accurately mapping the search space as it leaves large undefined gaps between the points. Any approximation of this search space could be highly inaccurate due to the large number of degrees of freedom. If this space were to be searched for an optimum, the results would likely prove to be erroneous when tested on real-world devices.

2.3 Generative AI

The interpolation of the search space between real-world data points is pivotal for accurately approximating the optimization function. Since neural networks are known for their high interpolation capability [33], this task is well performed by generative AI models [34]. These are adept at learning the search space representation from the available real-world data. Once trained, they can generate the missing points, which can then be used to calculate the value of the optimization function and map the search space with greater precision.

To facilitate training, some parameters believed to have minimal influence or relevance to the optimization function may be omitted. This helps reduce the dimensionality of the search space. Generative AI is capable of approximating all 3 components of the dataset - static, dynamic and stochastic.

2.3.1 Evaluation

Evaluating the performance of generative models is a research field of its own. In [35] Salimans et al. introduce the Inception score which is calculated through the use of an image classification model called Inception. This model is used to acquire a conditional label distribution. A good generated sample has a low entropy of the distribution. The authors state that it is a proficient validation metric with a high correlation to human judgment. This approach was further improved by Heusel et al. in [36] with the introduction of the Fréchet Inception Distance (FID). This new approach compares the distributions of the original training dataset and the generated image dataset to better evaluate the performance of a generative model.

Other evaluation methods may be used to compare generated samples with training images as well as approaches specific to the studied field as expanded further in Chapter 6.

2.3.2 Applications

Generative AI is one of the most discussed machine learning tasks today [37]. The rise and growth of text-conditioned image generation models in the most recent years is noticeable both in research and in pop culture. Many paid services like Midjourney [3] or Dalle [4] offer the use of image generation models to users via a subscription-based system. This approach helped spread the technology to the general public by offering it as an easy-to-use service. For more advanced users and researchers, the biggest open-source solution Stable Diffusion [38] is of big interest.

Generative AI has also been successfully used to augment training datasets. A review of such use in medical data was written in 2023 by Kebaili et al. [39]. The paper showcases the use of GANs, VAEs and most lately Diffusion models in enlarging the datasets of medical images such as CT, MR, ultrasound and others. It also shows the quick rise of Diffusion models in this area in the year 2022. Alsafadi et al. in [40] research the use of generative models in enhancing datasets related to nuclear engineering problems. They showcase the usefulness of increasing dataset size in fields where data production and gathering can be very expensive.

2.4 Transfer learning

Another solution to small datasets other than data generation is the use of transfer learning. Machine Learning (ML) models are generally trained and tested on data from the same distribution. This approach is changed in transfer learning by initially training on different data before transferring the acquired learned parameter weights to the required domain. This method can save training time while aiding in overcoming insufficient dataset size related issues. Transfer learning can be used in conjunction with image generation to increase the dataset size as the generator model first needs to learn the distribution of the limited dataset. The training of the generative model is therefore limited by the same problem it is intended to overcome. A survey of successful incorporation of transfer learning into machine learning processes was written by Weiss et al. [41] and offers further insight into the technique.

Diffusion probabilistic models

Diffusion models, a novel approach to image generation, were first introduced in [1] by Sohl-Dickstein et al. The fundamental concept, inspired by non-equilibrium thermodynamics, is the systematic and progressive transformation of the original data distribution through a forward diffusion process until it morphs into a known distribution, such as a Gaussian distribution. This process can be visualized as gradually adding noise to the data until the original structure is obliterated and a simple, known structure remains.

Once this forward diffusion process is complete, a model is trained to perform the reverse diffusion process. This reverse model learns to remove the noise and restore the original structure of the data. This approach allows the generating of new samples from the training data distribution. This is achieved by applying the learned reverse diffusion process to samples drawn from the simple known distribution (e.g., Gaussian noise data).

In this chapter, the mechanics of Denoising Diffusion Probabilistic Models (DDPMs) are explained. It explores how these models learn to reverse the diffusion process and how they can be used to generate new samples that resemble the training data. Furthermore, advancements in the field that have led to improvements in the performance and efficiency of these models are discussed. These enhancements have broadened the applicability of DDPMs and have opened up new avenues for their use in various domains.

3.1 Denoising Diffusion Probabilistic Model (DDPM)

The concept of diffusion models, first coined by Sohl-Dickstein et al. in [1], has since been recognized as a powerful tool in the realm of image generation. This was further substantiated by the research conducted by Ho et al., which established these models as the leading image generators [2].

The process of diffusion begins with the construction of a fixed Markov chain. This chain serves as the backbone of the model, guiding the transformation of the data. The transformation process involves the gradual infusion of Gaussian noise into the images. This noise follows a specific variance schedule, denoted as β_1, \dots, β_T [2]:

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

This variance schedule plays a crucial role in the diffusion process. It dictates the degree and pace of the noise infusion, ensuring that the transformation is both systematic and controlled. As the noise is gradually introduced, the original data distribution is slowly eroded and replaced with the Gaussian distribution.

The process of image generation using diffusion models involves a subsequent step where a model, typically a variant of the U-Net architecture [42], is trained to perform the reverse diffusion process. This model takes as input a partially noised image, denoted as x_t , and the number of remaining noise steps to be removed, represented as t . The output of this model is an image, x_{t-1} which has $t - 1$ noise steps remaining to be removed.

The objective function for this model is most commonly defined as the Mean Squared Error (MSE) between the removed noise, $\tilde{\epsilon}_t$ and the actual noise ϵ_t , that was added during step t in the forward diffusion process. This is calculated over a batch of size B , as shown in the following equation:

$$MSE = \frac{1}{B} \sum_{j=1}^B (\epsilon_{t_j} - \tilde{\epsilon}_{t_j})^2$$

When defined in this manner, the resulting model can be utilized in a process known as sampling to generate new images from the target distribution. This process is detailed in pseudocode 1. In this context, $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^t \alpha_s$.

Algorithm 1 Sampling

```

1:  $x_T \sim \mathcal{N}(0, I)$ 
2: for  $t = T, \dots, 1$  do
3:   if  $t > 1$  then
4:      $z \sim \mathcal{N}(0, I)$ 
5:   else
6:      $z = 0$ 
7:   end if
8:    $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}}\epsilon_\theta(x_t, t)) + \sigma_t z$ 
9: end for
10: return  $x_0$ 

```

3.2 Denoising Diffusion Implicit Model (DDIM)

While DDPMs have proven to be effective, one of their limitations is the slow sampling speed. The generation of an image necessitates the simulation of a complete Markov chain of the diffusion process, which often comprises up to 1000 noising steps. This can be computationally intensive and time-consuming.

In an attempt to address this issue, Song et al. introduced a novel non-Markovian generative process in their work [43]. This process shares the same training objective as DDPMs, which means that the same models can be utilized for Denoising Diffusion Implicit Model (DDIM) generation without necessitating any additional training. The process is deterministic, allowing for the execution of only a subset of the steps in the reverse diffusion process, with the remaining steps being interpolated. This results in a significant speedup, with the authors noting an improvement of around 10 to 50 times in wall clock time when compared to DDPM sampling.

Furthermore, this approach introduces a trade-off between sample quality and generation time. Taking more steps can yield higher-quality results but at the cost of increased computation time. This enhancement broadens the applicability of diffusion models, making them suitable for real-time solutions and enabling the generation of large datasets within a feasible timeframe.

The proposed diffusion process adheres to the following distribution:

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(\sqrt{\alpha_{t-1}}x_0 + \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \frac{x_t - \sqrt{\alpha_t}x_0}{\sqrt{1 - \alpha_t}}, \sigma_t^2 I)$$

A sample x_{t-1} can be generated from a sample x_t using the following equation:

$$x_{t-1} = \underbrace{\sqrt{\alpha_{t-1}} \left(\frac{x_t - \sqrt{1 - \alpha_t} \epsilon_\theta^{(t)}(x_t)}{\sqrt{\alpha_t}} \right)}_{\text{predicted } x_0} + \underbrace{\sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \epsilon_\theta^{(t)}(x_t)}_{\text{direction pointing to } x_t} + \underbrace{\sigma_t \epsilon_t}_{\text{random noise}}$$

Here, $\epsilon_t \sim \mathcal{N}(0, I)$ represents standard Gaussian noise that is independent of x_t , and $\epsilon_\theta^{(t)}$ is the output of the model given timestep t . It is also important to note that $\alpha_0 := 1$.

While the generative process was originally designed to have T steps, mirroring the forward diffusion process, it is possible to redefine the forward process using a sequence τ of noising steps of length S . This results in $q(x_{\tau_i} | x_0) = \mathcal{N}(\sqrt{\alpha_{\tau_i}} x_0, (1 - \alpha_{\tau_i}) I)$. When S is significantly smaller than T , the sampling process becomes much faster as it requires fewer steps, while still using the same model [43].

3.3 Diffusion-based Autoencoders (DiffAE)

Diffusion-based generative models have surpassed Generative Adversarial Networks (GANs) in terms of the quality of sampled images. With the advent of DDIMs as proposed by Song et al. [43], these models have also achieved parity with GANs in terms of sampling speed. However, one limitation of diffusion models is that their latent space lacks semantic meaning, which restricts their utility for representation learning and other tasks.

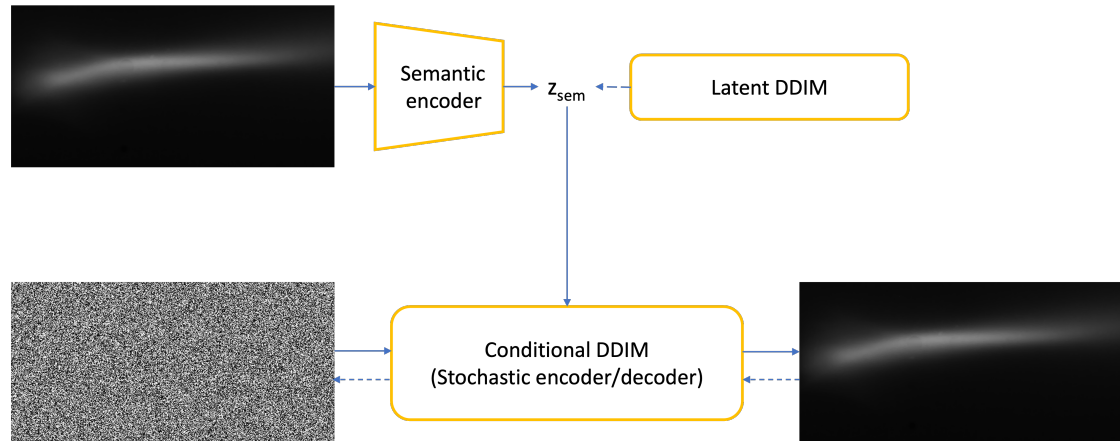
In an attempt to overcome this limitation, Preechakul et al. introduced the concept of a Diffusion Autoencoder (DiffAE) in their work [44]. The central idea behind DiffAE is the incorporation of a separate semantic encoder that conditions the base DDIM. This transforms the DDIM into a stochastic decoder, with the semantic component being supplied externally. Interestingly, this stochastic DDIM can also function as an encoder, thanks to the ability of DDIMs to reverse the learned sampling process. This dual functionality enables the creation of meaningful latent representations of the input data using the stochastic and semantic encoder.

However, this architecture does not inherently support the generation of new data. It can only produce different stochastic variations of an image, as the DDIM needs to be conditioned by a latent semantic representation of an image. To address this, the authors introduced a third component to the architecture: a DDIM capable of performing sampling in the semantic latent space. During the sampling of new data, only the latent and stochastic DDIM are used, bypassing the semantic encoder. The complete architecture of the DiffAE is illustrated in Figure 3.1.

3.4 Classifier-free guidance

To achieve more nuanced guidance during the image generation process, it becomes necessary to introduce a condition. This condition could be a text prompt describing the desired image, as is commonly employed in most commercially used generative AI models [3, 45, 38]. However, for the simulation of electron acceleration with lasers, as presented in this thesis, a vector containing the experimental settings proves to be the most suitable condition. This vector, in conjunction with the variable t denoting the timestep of the diffusion process, serves to condition the model.

Ho et al. propose an innovative approach termed “classifier-free guidance” in their work [46]. This method enhances the quality of sampling by controlling mode coverage, which ensures an evenly spaced distribution of the data in the learned latent space, and sample fidelity, which measures how closely the generated data points map to the real-world representation. This is achieved by intermittently removing the condition during parts of the training process, thereby compelling the model to learn the unconditional representation of the dataset. This typically applies to a small fraction of the training batches, such as 10%.



■ **Figure 3.1** Overview of the architecture of the DiffAE model from [44]

During the sampling process, the algorithm is provided with a condition, but it generates two samples - one conditional and one unconditional. The final generated data point is produced by interpolating between these two samples. The influence of the conditional sample can be modulated by a variable termed *cfg-scale*, which controls the weight of the conditional sample during interpolation. This approach offers a balance between the specificity of conditional sampling and the diversity of unconditional sampling, thereby enhancing the overall quality and utility of the generated data.

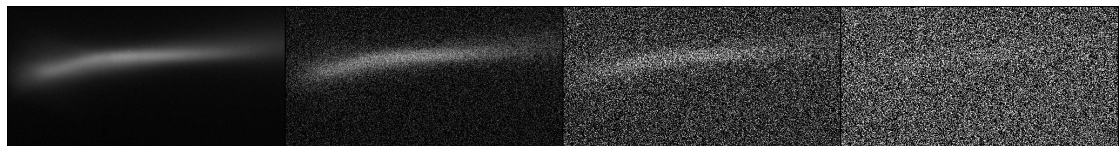
3.5 Inverse heat dissipation

The generative process of traditional diffusion models primarily focuses on the removal of per-pixel noise, without taking into account the overall structure of the image. This means that the influence of neighboring pixels during the noising process is not considered, which could potentially limit the quality of the generated images.

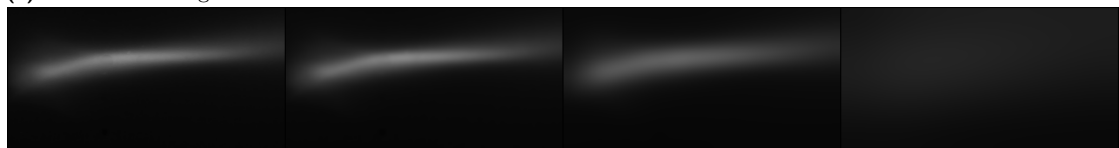
An alternative approach that addresses this limitation by considering the structure within the image during the noising process was proposed by Rissanen et al. in their work [47]. In this novel method, the authors implement a forward heat equation with constant additive noise for the forward diffusion process. This approach can be visualized as an iterative “melting” process, where the input image gradually loses its distinct features until it reaches a state of uniformity.

The sampling process then reverses this “melting” by progressively removing the blur, while also introducing a small amount of random noise. This addition of random noise ensures that the image generation process is non-deterministic, thereby allowing for the creation of diverse and unique images. This process, and its distinction from standard diffusion, is illustrated in Figure 3.2.

By considering the influence of neighboring pixels, this method allows for a more holistic and context-aware generation of images, potentially leading to improved results in terms of both image quality and diversity.



(a) Gaussian noising



(b) Blurring

■ **Figure 3.2** A comparison between Gaussian noising and blurring as introduced in [47].

Physics informed machine learning

The training of machine learning models is often very time-consuming and it can be difficult to successfully train a model. A large amount of data is needed to train a model simulating complex high-dimensional problems. It is often the case with such problems that data collection is expensive or time-consuming leading to smaller datasets. The collected data can also be noisy and it may not always be straightforward to remove such noise in preprocessing. In such cases, it is beneficial to introduce some constraints to the models to better guide the training process. These constraints can be in the form of physical laws which are to be obeyed. This can help with accuracy and efficiency by shortening the time needed for training and improving the performance of the resulting model.

4.1 Implementation of physics into the training process

Incorporating physics into ML processes enhances the predictive accuracy and generalizability of models. The integration of physics can be approached through various methodologies, as outlined in the comprehensive review by Karniadakis et al. (2021) [48]. This section elucidates the implementation strategies and their implications for the training process.

4.1.1 Observational bias

The concept of observational bias is introduced through the training dataset itself. It encompasses any recurring pattern or element that embodies the effects of physical constraints. This form of bias is inherently passive, relying on the assumption that the model will infer the underlying physical principles through exposure to a sufficiently large and representative dataset. While this approach aligns with the data-driven nature of ML, it may necessitate an expanded dataset to achieve its full potential. The presence of observational bias can be a direct consequence of physical laws influencing the data or a deliberate enhancement of the dataset in an attempt to describe such effects. [48]

4.1.2 Inductive bias

Inductive bias is implemented within the neural network's architecture. By design, the network is constrained to produce outputs that are consistent with the physical laws, thereby ensuring compliance with these rules. This method offers precision at the cost of reduced model flexibility. It is particularly applicable to aspects of physics that can be seamlessly translated into architectural constraints. However, the complexity of certain physical phenomena poses challenges to

this approach, as a comprehensive understanding is required for effective implementation. Employing inductive bias on incompletely understood physical processes risks embedding erroneous assumptions, potentially hindering the model's objectives by constraining it inappropriately. [48]

4.1.3 Learning bias

Learning bias is introduced during the network's training phase. It involves the modification of the existing loss function or the introduction of a novel one, steering the model's convergence towards solutions that are congruent with the physical laws indicated by these loss functions. This strategy is the most flexible, allowing for the incorporation of a wide array of physical principles, including those that are not entirely understood. The degree of bias can be finely tuned, providing control over its influence on the model's learning trajectory. [48]

4.1.4 Hybrid approaches

A hybrid methodology fuses multiple biases to leverage their respective strengths. This is particularly beneficial when dealing with multifaceted physical systems, where certain components may be reduced to inductive bias while others, less comprehensible, are better suited for learning bias. Additionally, the combination of observational bias with other forms is advantageous in scenarios involving extensive datasets. [48]

4.1.5 Implementation for electron acceleration

In this thesis, the learning bias was selected due to the incomplete comprehension of the introduced physical constraints. The scarcity of data precluded the effective utilization of observational bias, and the diffusion model's emphasis on noise reduction complicated the application of inductive bias. The subsequent sections will delve into the primary sources of physical information that informed the learning process.

4.2 Calculation of electron beam spectra

The primary focus of the investigation into electron acceleration is centered around the calculation of electron beam spectra derived from the electron beam's trace on the Lanex screen. By examining the images of this screen, the energy of the accelerated electrons based on their displacement from the initial electron beam position can be deduced. Electrons situated farther from the beam's origin possess diminished kinetic energy, a consequence of their trajectories being more significantly altered by the electromagnet compared to their high-energy counterparts. The precise energy values of these electrons can be gathered by employing a deflection curve tailored to the experimental configuration. The amount of accelerated electrons can be deduced from the intensity values of the image. The generation of the final spectra can then be achieved by aggregating the image along the horizontal axis to gather the particle intensity at specific distances. This data, in conjunction with the deflection curve and the pixel-to-millimeter conversion factor, facilitates the computation of the electrons' energy.

Electron beam spectra form the main analytical focus for researchers. It conveys information on the accelerated particles' energy whose maximization is the main focus and therefore it is reasonable to convey this information in the diffusion model. In the physics-informed version of the final model, the spectra are used during the loss computation phase.

This approach was used to guide the model toward noise removal which would preserve the spectra from the training data point. It was believed that doing this would ensure both faster convergence during training and more precise results as the model would more accurately adhere to the deflection curve.

4.3 Penalisation based on electron beam position

An important element in the dataset is the position of the electron beam. This point (estimated by a single pixel) marks the position of accelerated electrons if no magnet is inserted into the aperture. Since this magnet pulls the particles toward the right side of the image, it should be impossible to observe any particles to the left of the beam-pointing pixel.

However, the inherent noise within the dataset occasionally results in non-zero values in regions situated to the left of the beam-pointing pixel in the training images. These anomalies are attributed to either noise interference or unidentified phenomena that fall outside the simulation scope of the model. Therefore it was believed that penalizing the model for keeping non-zero values in such positions would lead to better and more meaningful results.

This bias could be also introduced in the form of an inductive bias by allowing the network to only generate pixels to the right of the electron beam. This approach was deemed not flexible enough, as the position of the beam is an approximation and therefore may be slightly misleading. Furthermore, electrons generated near the original beam position carry exponentially more energy in relation to their distance to this point. It is therefore reasonable to focus on these areas and include them not only as the edges of the generated image. Generation of the rest of the image may also help explain why electrons appear in these physically non-sensical areas and help explain other phenomena happening in the background of electron acceleration.

Optimization

Laser-plasma particle acceleration is a process influenced by many parameters, sensitive to their changes. Finding an optimum for these parameters is a difficult task that requires tuning a plethora of settings of both the laser and the plasma. Some of these can be approximative given their fluctuating nature making the measurements noisy and difficult to predict. While the influence of each parameter is conceptually known and explained, their mutual influence is difficult to understand and therefore predict. [49]

Given these obstacles, the path toward an optimal set of parameters can be long, unpredictable and filled with missteps even for highly knowledgeable physicists. The optimization goal is usually tied to some quality of the generated electron beam. This can be either its stability or the energy of accelerated particles. The former can be analyzed through a series of images of the Lanex screen, where an unstable beam might not appear during every shot of the laser or change its position within the screen. The energy of the particles is measured with a magnet present in the aperture through a deflection curve and its influence on the position of the electrons on the Lanex.

In this chapter different approaches to the optimization of laser-plasma particle acceleration are discussed as well as their successful implementations.

5.1 Bayesian optimization

Bayesian optimization serves as a surrogate model approach to approximate the underlying problem, typically characterized by a computationally expensive-to-evaluate function, with a more tractable surrogate function. This surrogate function is leveraged to efficiently explore the parameter space and identify sets of parameters that yield favorable outcomes in the original problem. It operates under the assumption that the original function is a realization from a Gaussian process, thereby constructing a posterior distribution informed by new observations [50]. These observations are systematically acquired by evaluating the original function. To optimize the search process while minimizing the number of function evaluations, the algorithm selects parameters for new observations by maximizing acquisition functions such as the expected improvement (EI) [51] or the Gaussian process upper confidence bound (UCB) [52].

In a letter by Jalas et al. [49], Bayesian optimization was applied to a laser-plasma accelerator to enhance beam quality. The devised objective function aimed to maximize beam charge while concurrently minimizing energy spread, favoring the generation of intense, uniform-energy beams. Such beam characteristics are particularly advantageous for applications like free-electron lasers [53], which necessitate electrons with narrowly distributed energies.

The optimization process monitored parameters including gas density, N_2 concentration, laser

energy, and laser focus position. Initially, the optimization commenced with PIC simulations, using a set of five randomly sampled points as a foundation. Subsequently, the Bayesian optimization algorithm suggested new parameters and PIC simulations were employed to obtain additional samples. The findings from this iterative process demonstrated that, under certain conditions, the generation of beams with minimal energy spread is feasible. When the algorithm was subsequently applied to an operational particle accelerator, it converged to a configuration yielding beams with subpercent energy spread within 45 minutes, starting from arbitrary parameter values.

However, the authors noted that the laser parameters exhibited instability, rendering the measurements somewhat unreliable. The model's precision was further compromised by fluctuations in other unmonitored parameters. To counteract these effects and enhance beam stability, the researchers averaged 20 measurements with variable parameters instead of relying on a single measurement. The most frequently occurring objective value among these measurements was then selected, significantly bolstering the stability of the resultant beams. The authors concluded that the synergy between Bayesian optimization, PIC simulations, and real-time experimental data constitutes a formidable toolkit for optimizing laser-plasma particle accelerators.

Complementary findings were reported by Shaloo et al. [14], where the optimization initially focused on maximizing the total count of accelerated electrons, followed by optimizing for the yield of betatron X-rays in a subsequent phase. Echoing the observations of Jalas et al., they acknowledged the detrimental effects of laser performance drift and other experimental variables. Despite these challenges, both optimization strategies successfully converged to optimal solutions within 40 iterations (laser shots).

The study emphasizes that the selection of the acquisition function significantly influences the resulting optimal settings. Moreover, the researchers evaluated the surrogate model generated by the Bayesian optimization process. A discernible correlation between two of the parameters was identified, and their impact was further scrutinized through PIC simulations, illustrating the efficacy of machine learning in elucidating intricate physical phenomena.

An approach similar to that of the authors of this paper is expected to be used on the model developed in this thesis. The model will be used instead of the real experiments to allow more iterations of (simulated) laser shots at much lower costs. The achieved optimized parameters will then be used in the next iteration of real laser experiments. The data gathered from these experiments can then be used in retraining the model and repeating the optimization process.

5.2 Reinforcement learning

At the heart of Reinforcement Learning (RL) lies the dynamic interaction between an agent and its environment. The agent, through a series of actions, navigates the environment, each action yielding a reward. These actions not only confer immediate rewards but also influence the environment, altering the landscape of potential future actions and their associated rewards. The defining features of RL are the trial-and-error learning process and the concept of delayed rewards, which together facilitate the discovery of optimal strategies over time [54].

In the work of Capuano et al. [55], RL was employed as an optimization tool to fine-tune laser parameters for optimizing laser pulse shapes. The researchers developed a semi-physical model that emulates the laser's operational dynamics. Within this simulated environment, the agent was granted the ability to adjust three critical settings identified as having the most significant influence on pulse shape. Following each adjustment, the environment provides feedback in the form of a new state and a reward, calculated based on the proximity of the pulse shape to a predefined target. To safeguard the machinery, the agent's adjustments are constrained, prohibiting drastic changes in a single iteration. The study proposed two distinct models, differentiated by their reward functions. The first model serves as a proof of concept, demonstrating the feasibility of using RL for controlling the temporal shape of laser pulses. The second model, more informed by underlying physics, employs a reward function tailored to specific task objectives.

The results of this research were promising, showcasing an RL agent capable of manipulating laser parameters to achieve target outcomes efficiently and with minimal steps. The agent's robustness against random initial states was also demonstrated, even under the restrictions imposed on parameter adjustments to ensure the safety of the machinery. The authors advocate for the validation of their approach through experiments with an actual laser system, as their study was based on a semi-physical simulation rather than a real-world environment.

Architecture and implementation

This chapter covers the specifics of the architecture of the developed and tested models, processes used in dataset preparation and the sampling algorithm details. Key features of the network are described and visualized. For the implementation of the neural network the Pytorch library was used [56]. The base architecture as well as DDPM sampling are an enhanced version of the implementation by Dominic Rampas [57].

6.1 Data preparation pipeline

The dataset provided by the physics team required several modifications to make it suitable for training a generative AI model. The dataset is composed of 22 folders, each containing between 31 and 110 shots. The team at Eli utilized a Matlab script to preprocess the dataset. To ensure compatibility with the rest of the project, this script was refactored and rewritten in Python. This transition necessitated several adjustments due to differences between Matlab and Python, such as the indexing starting at 1 in Matlab and 0 in Python, and the differing behavior of the round function.

The received dataset exhibited a high level of noise, resulting in grainy images. To mitigate this, the data was preprocessed using a median filter with a kernel size of 5. A larger kernel did not appear to significantly improve noise removal. Additionally, the images were taken by camera with a damaged sensor making parts of the images always contain maximum value. These parts were instead set to 0 as is usually the background value. Fortunately, these errors never appear near the electron beams and therefore such treatment is sufficient.

The Lanex screen, which is hit by accelerated electrons (as seen in the Alpha campaign setup diagram in Figure 1.1), has black dots arranged in a cross shape to facilitate distance calculations. These dots obscured the image and needed to be removed. In the original script, the location of each dot was hardcoded, which is not ideal as the position of the dots can vary between experiments due to the potential movement of the camera. This process was automated using a dot-detection algorithm. The image is first converted to binary using adaptive thresholding. The main parameters are *blockSize*, which controls the size of the neighborhood used to calculate the segmentation threshold, and *C*, a constant subtracted from the mean of the neighborhood to obtain the final threshold value. After experimentation, these parameters were set to 51 and 4 respectively. The dots were then detected using the circle hough transform [58]. The OpenCV [59] implementation was used with the following parameters: *minDist=20*, *param1=50*, *param2=7*, *minRadius=0*, *maxRadius=10*. This resulted in more detected dark dots in the image than actually present, which is preferable to missing any. The black dots were then removed by interpolating along the *x* axis, meaning that the “removal” of non-existent dots had minimal

effect on the resulting image.

In addition to the data with the magnet in the aperture, which is used to measure the energy of the electrons, there are corresponding datasets without the magnet used to acquire the position of the electron beam. This position is then used as a reference point when calculating the energy of the accelerated particles. The position of the electron beam was estimated by summing all images of a given experiment and then searching for the position of the maximum in the summed image. This point of highest intensity is believed to be the peak of the electron beam.

The next step in image preprocessing is cropping. The images were cropped to a height of 256 pixels and width of 512 pixels, with the electron pointing pixel fixed at coordinates [128, 62] (vertical and horizontal position). This position was selected because the vertical axis is similarly important, and on the horizontal axis, the electrons are always being pulled to the right of the default electron beam position. However, this image size is quite large for training a convolutional network and would require more VRAM than was available for this project. Cropping the images to a smaller size of 64 by 128 pixels resulted in many of the training data being almost entirely black and therefore of little use. As a result, the images were resized using bilinear interpolation before being inputted into the model. At the end of the sampling process used for image generation, the images are resized back to the original size. Due to the gradient-like nature of most of the images, this process of back-and-forth interpolation does not result in much loss in the fine detail of the images.

The process of the dataset pipeline can be seen in Figure 6.1.

Together with the image data the dataset also contained an Excel file with parameters for each experiment. These were as follows:

- **E** - energy of the laser
- **P** - pressure in the gas chamber
- **gas** - type of gas (Nitrogen or Helium/Nitrogen mixture)
- **ms** - acquisition time of the camera
- **gain** - gain setting on the camera

In the process of training the model, only a subset of the parameters, specifically **E**, **P**, and **ms**, were incorporated into the conditional vector. The rationale behind this selection was based on the relevance and variability of these parameters in the context of the experiments.

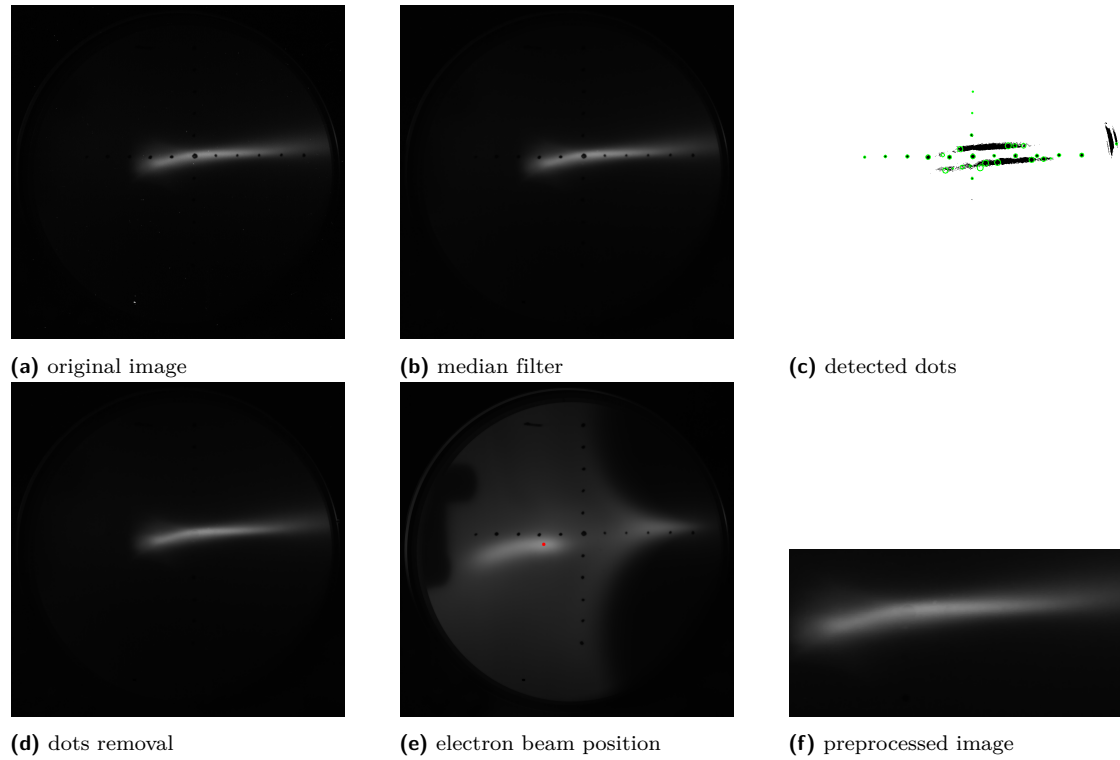
The parameter **gas**, which denotes the type of gas used in the experiments, was deliberately excluded from the conditional vector. This decision was driven by the fact that only two experiments were conducted using a gas mixture, while the rest employed Nitrogen. Consequently, the **gas** parameter exhibited very little variation across the dataset, rendering it less informative for the purpose of training the model.

Furthermore, both **gain** and **ms** are parameters related to the camera settings, and they influence the brightness of the captured images. However, the **gain** parameter was not included in the conditional vector. The influence of the **gain** setting on the image brightness is deterministic, meaning it follows a predictable and consistent pattern. Therefore, its effect can be accounted for by subtracting it from the images during the preprocessing stage. This allows the model to focus on learning from the more variable and informative aspects of the data, potentially improving its performance and generalization ability.

The data was collected using a Basler aca2040-25gm camera. According to the official documentation [60], the formula for calculating gain is:

$$\text{Gain} = 20 * \log_{10}(\text{GainRaw}/32)$$

Here, *GainRaw* is the value set within the Basler Pylon Viewer through which camera settings were adjusted. After recalculating the value of *Gain* from decibels (dB) to linear units, the gain



■ **Figure 6.1** Preprocessing pipeline.

value is used to adjust the brightness of the image. However, it was experimentally found that subtracting the gain value from the dataset causes the network to perform poorly during training. It is assumed to be caused by weak signal from dataset features. Therefore, the gain value was left as is. This may result in some inaccuracies when accessing the intensity of the generated samples. However, this inaccuracy was deemed unsubstantial and a better alternative to poor training performance.

6.2 Diffusion probabilistic model

The input image is processed through a U-Net architecture [42], which is composed of three downsampling and upsampling blocks. The initial processing stage involves a double-convolution block, which applies a 2D convolutional layer, group normalization, and a GELU activation function twice in succession. This block's structure is depicted in Figure 6.2a.

Following this, the first downsampling block is applied, which consists of a 2D maxpooling layer and two subsequent double-convolution blocks. The schematic representation of the Down block can be found in Figure 6.2b.

In addition to images, the network processes the settings vector and the noising step number. The settings vector, denoted as y , is processed through a block of layers, as described in Figure 6.2c, which includes batch normalization, a linear layer, and a SiLU activation. The noising step number, denoted as t , is encoded using positional encoding to provide the model with more nuanced information about the noise step beyond a single integer value. The tensors y and t are combined and serve as inputs to the network's Down and Up blocks, where they undergo further processing through a SiLU activation and a linear layer before being added to the tensors derived from the image data. This approach equips the model with both conditional information and the

remaining number of noise steps to be eliminated.

At the network’s core, there are three double-convolution blocks, succeeded by the upsampling phase. The Up block mirrors the structure of the Down block, except for an upsampling layer replacing the maxpooling layer to enlarge the tensor dimensions. Layers are concatenated by a skip connection, utilizing the output from the corresponding Down block. A comprehensive illustration of the Up block is provided in Figure 6.2d.

Each Down and Up block is followed by a SelfAttention [61] block. Initially, the input is reshaped to conform to the data ordering expected by the MultiheadAttention layer and is then normalized using layer normalization. The output is subsequently combined with the original input, followed by a sequence of layer normalization, linear transformation, GELU activation, and another linear transformation. Prior to reshaping the output to maintain consistency with the network’s data ordering, the output of the MultiheadAttention layer is added once more. The SelfAttention block’s structure is detailed in Figure 6.2e.

The complete architecture of the model is visualized in Figure 6.3.

6.2.1 Training loop

The training process was executed on an NVIDIA A100 graphics card, utilizing the AdamW optimizer [62] in conjunction with the cosine annealing learning rate scheduler [63]. The diffusion process was initialized to facilitate image sampling during training, enabling progress visualization and the execution of the forward diffusion process on the training dataset. The noising schedule was configured by setting the start and end β variables, which control the amount of noise introduced at each step. The implementation was changed from the original linear noising schedule to a cosine schedule for increased sampling performance. A visual representation of the forward diffusion process is provided in Figure 6.4.

The model underwent training for 300 epochs, with a batch size set to 8 images. This batch size was heavily constrained by the memory requirements for training, as any increase would exceed the maximum VRAM capacity of the graphics card.

Prior to network entry, the image dataset underwent several transformations. All images were converted into tensors, resized to dimensions of 64 by 128 pixels (height and width), and normalized to a mean of 0.5 and a variance of 0.5. During this normalization process, images lacking distinctive features and only containing values close to zero were transformed into highly noisy data. To counteract this, a maximum-value “fingerprint” was assigned to all images in the upper-left corner prior to normalization.

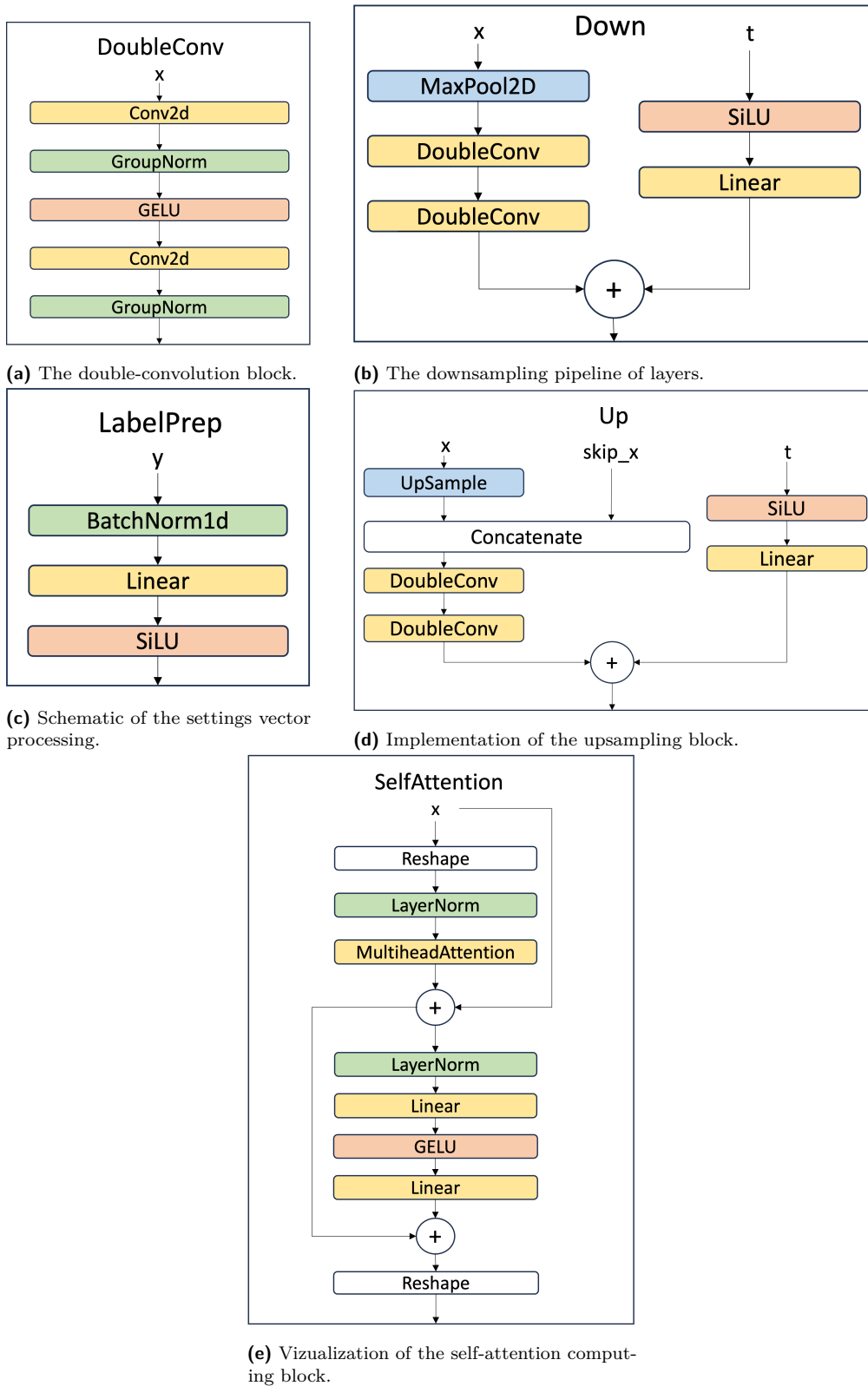
Each image within a batch was assigned its corresponding settings vector y and a random noise step value t . The image was then subjected to the forward diffusion process for t steps before being inputted into the model for denoising to $t - 1$ steps. The loss was computed as the mean squared error between the noise removed (as predicted by the model) and the noise added during this step of the forward process.

6.2.2 Sampling algorithm

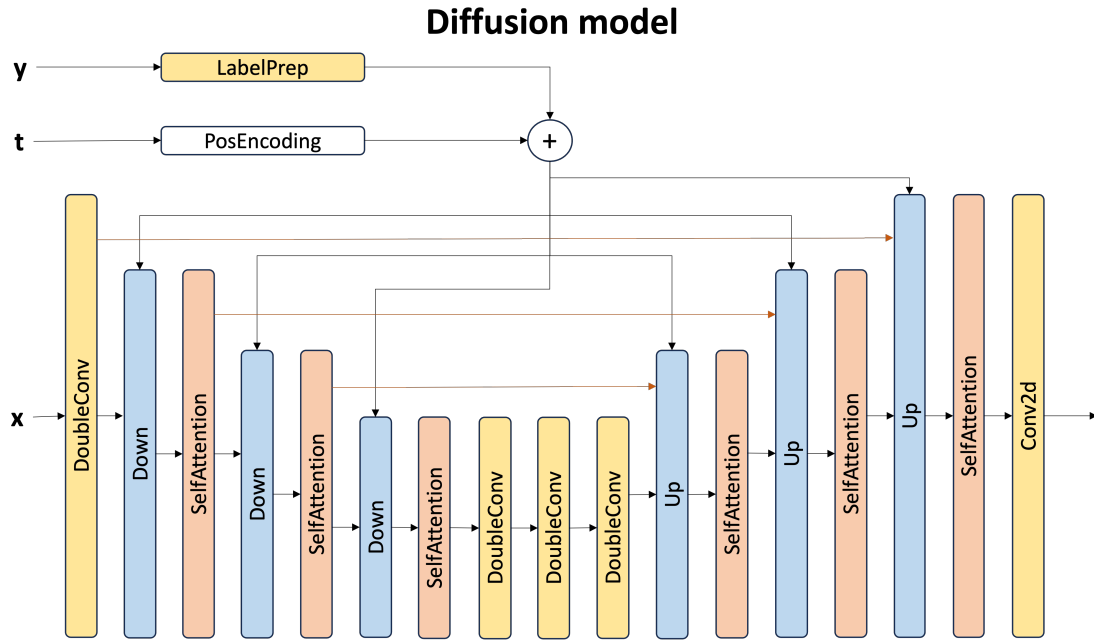
The original DDPMs sampling sequence, designed for conditional generation, commences by generating an image composed solely of Gaussian noise. This image, along with the maximum noise step value and the settings vector, is fed into the model. Following the removal of a noise layer by the model, the remaining noise step number is decremented, and the image is re-inputted into the model. This iterative process continues until all noise steps have been executed.

Upon completion of the sampling process, the output image is reshaped to the desired dimensions. In the final implementation, this step is crucial for generating the image in its original size, as the training dataset was previously rescaled to minimize computational demands.

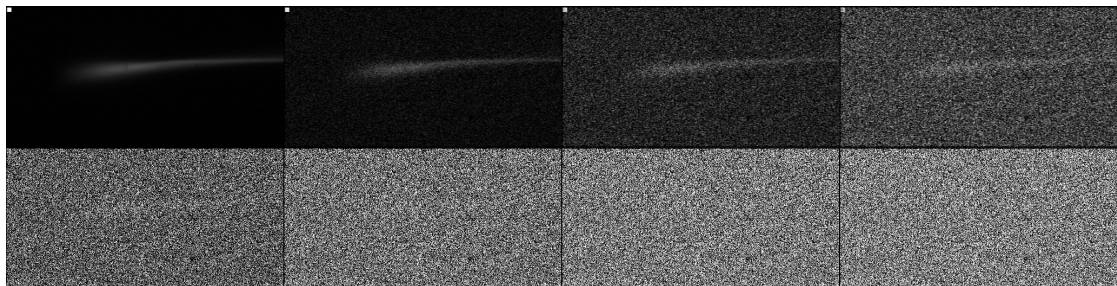
The enhanced DDIM sampling sequence, as described in [43], was implemented utilizing the source code for the diffusion autoencoder [44]. This implementation leverages the underlying



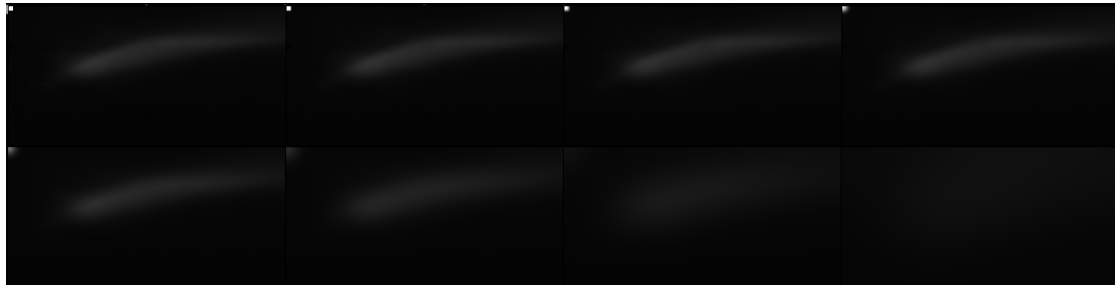
■ **Figure 6.2** Schematics of the building blocks of the diffusion model’s final architecture. Arrows suggest the input/output of tensors. Visualized in green are normalization layers, red are activation layers, blue are layers changing the scale of tensors and white are layers with no learnable parameters. Other layers with learnable parameters are in yellow.



■ **Figure 6.3** Schematic of the full model architecture. Arrows suggest the input/output of data; red arrows symbolize skip connections. Tensor x contains image data and tensors t and y contain the noise and settings respectively. Individual blocks are described in Figure 6.2.



■ **Figure 6.4** Effect of the noising process. Images from left to right are in 100 noise steps increments from 0 to 700.



■ **Figure 6.5** The effect of the inverse heat dissipation blurring. The input image in the top left is processed by 100 noise steps in each of the following figures going from left to right, top to bottom.

DDPM noising schedule to derive a new schedule with a reduced number of steps. Furthermore, it offers the flexibility to distribute the new steps non-uniformly, enabling a more precise focus on specific segments of the denoising process.

Both sampling methodologies employ classifier-free guidance improvement by executing both unconditional and conditional generation at each step. The final sample is produced by interpolating the results from these two approaches.

6.2.3 Inverse heat dissipation

The implementation of Inverse Heat Dissipation (IHD) was based on the official code released by the authors [47]. Analogous to the Gaussian sampling, a beta schedule is initially created to delineate the quantity of noise introduced at each step. During the sampling phase, the initial fully noised image is generated by selecting a random image from the training dataset and applying the full forward blurring process. This image is then processed through the noise-removing network for a number of steps equivalent to the total noise steps. At each denoising step, a small quantity of random Gaussian noise is added to the image, rendering the sampling process non-deterministic and enabling the sampling of the full distribution.

The full blur noising process is viewed in Figure 6.5.

6.2.4 Transfer learning

Transfer learning, a strategy that leverages pre-existing knowledge from a different model, was employed to expedite the training process. The source model chosen for this purpose was the U-Net model, which had been trained on ImageNet with a conditional vector by Karras et al. in [64].

Weights from the encoder and decoder blocks of the source model were selected and integrated into the corresponding Down and Up blocks of the target model. This process involved the transfer of weights to and from 2D convolutional layers. If the number of channels in the source and target were identical, the weights were directly copied. In cases where the source model had more channels, a random selection of channels, equal to the number of destination channels, was used. Conversely, if the destination layer had more channels than the source, random channels from the source were reused to populate the additional destination channels.

During the training phase, to preserve the transferred knowledge from the source model, layers with transferred weights were assigned a lower learning rate value. Specifically, layers in the bottommost DoubleConv blocks of the model were assigned a learning rate of $1e - 5$. The two Up and Down blocks closest to the bottom (Down2, Down3, Up1, and Up2) were assigned a learning rate of $1e - 4$. The remaining layers retained the learning rate of $1e - 3$, consistent with the rest of the learnable parameters.

The implementation of transfer learning resulted in an accelerated convergence towards meaningful sampling results, thereby reducing the overall training time for the model. This approach effectively leverages the power of pre-trained models to enhance the performance and efficiency of the target model.

6.3 Physics-informed loss penalization

The integration of physics-informed penalization into the model is controlled by one of the developed hyperparameters, which sets the noise step number at which the penalization is applied. During the initial stages of noise removal (specifically, the first half of the noising steps), the output generated does not closely resemble an image from the dataset. Consequently, it was deemed counterproductive to apply penalization at this stage. However, once the images begin to approximate the desired output, the methods delineated in Section 4 are employed. The noise step threshold for the application of these methods was set at one-tenth of the total number of noise steps.

6.3.1 Electron beam spectra simulation

The implementation of the electron beam spectra calculation is derived from a Matlab script provided by the scientists at Eli Beamlines. The first step in this process involves the elimination of background noise from the image. The value for this noise is determined by taking the median of three selected pixels from the image, located near the top right, bottom right, and bottom left corners. These regions are chosen as they rarely contain any accelerated particles, and thus, the values present are attributed to noise. Any pixel value below the calculated background noise is subsequently set to zero. Next, the distance of each pixel in the image from the electron-pointing pixel along the horizontal axis is calculated. This is achieved by multiplying the pixel distance by the pixel-to-millimeter ratio, which was determined during the conducted experiments to be 0.137mm per pixel. This value is based on the focal point of the lens mounted on the camera capturing the images and the distance between the Lanex screen and the camera's sensor. This calculation yields an array of distances in millimeters for each pixel from the electron-pointing. The loaded deflection curve is then utilized to convert the distance in millimeters to the energy of particles in MeV (mega electron-volt).

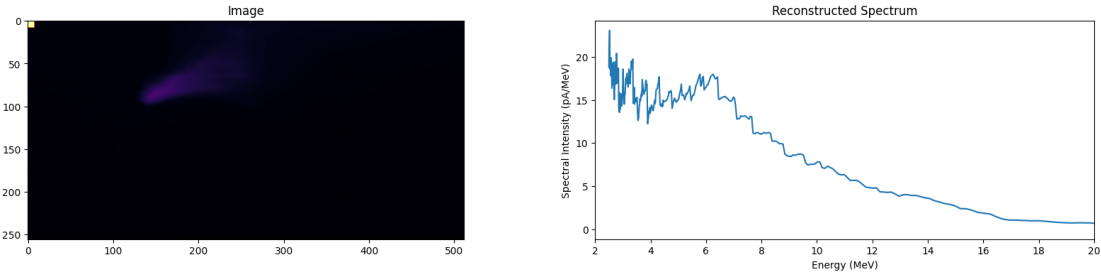
To obtain the spectrum, the horizontal sum of the image, termed the spectrum in millimeters, is computed. This spectrum contains information about the number of particles present at each given distance from the electron-pointing pixel. These values are then converted to MeV using the corresponding values of the loaded deflection curve from the previous step. Finally, the spectrum is calibrated by dividing it by the acquisition time in ms. This step is necessary as the camera allows light onto the sensor for the duration of acquisition, thereby enabling multiple waves of electrons to be observed on the Lanex during the acquisition of a single image.

A training image, along with its calculated electron beam spectrum, is visualized in Figure 6.6.

This information is incorporated into the training process by creating a secondary loss value. This value is obtained by calculating the mean squared error between the spectrum of an image denoised by the model and an image denoised by reverting the forward diffusion process. To ensure that this value meaningfully influences training, it is multiplied by a factor of 10 before being added to the total loss.

6.3.2 Electron beam pointing pixel

The implementation of the penalization strategy, as outlined in Section 4.3, involves a specific calculation related to the electron beam pointing pixel. This calculation is performed by summing



■ **Figure 6.6** This figure presents an image from the training dataset, accompanied by its corresponding spectrum calculation. The image is rendered in monochrome, with a colormap applied to enhance visual clarity and facilitate easier interpretation.

the values of all pixels located to the left of the electron beam-pointing pixel. It is important to note that the “fingerprint” in the top left corner of the image is excluded from this calculation. This exclusion ensures that the calculation accurately represents the relevant pixel data, without being skewed by the fingerprint values.

This approach to penalization, which is informed by the physical properties of the system under study, contributes to the robustness and accuracy of the model’s performance. This enhances the model’s ability to generate meaningful and reliable outputs.

6.4 Diffusion autoencoder

The implementation of the DiffAE as proposed in [44] was developed as suggested by the authors and adjusted for the implementation of the diffusion model as implemented in this thesis. The autoencoder is a work in progress and is not the main product of this thesis. It is however the most promising point of further development as discussed further in Chapter 8.

6.4.1 Architecture

The architecture of the model is composed of three integral components: the stochastic encoder/decoder, the semantic encoder, and the latent sampler. The stochastic encoder/decoder, which constitutes the largest submodel, shares its architecture with the diffusion model proposed earlier in this chapter.

The semantic encoder, on the other hand, aligns its design with the initial part of the diffusion model. It comprises four Down blocks, each succeeded by a SelfAttention block. This exceeds the three-block sequences observed in the previous model. However, the number of blocks is a tunable hyperparameter that directly influences the dimensionality of the semantic embedding. This semantic embedding serves a crucial role in conditioning the stochastic encoder/decoder.

The final component of the autoencoder is the latent space sampler. This is designed as a DDIM, consisting of a sequence of linear layers and normalization layers. The number of these layers also serves as a hyperparameter, providing additional flexibility in the model’s configuration.

The resulting implementation enables the model to deconstruct input data into its stochastic and semantic components, paving the way for in-depth analysis. The semantic vector is obtained through the semantic encoder, while the stochastic vector can be procured by executing a reverse sampling operation (a feature of DDIMs) with the stochastic encoder. This analysis aims to enhance the understanding of how parameters influence the experiment and identify the elements left unexplained in the stochastic part by these parameters.

6.4.2 Training loop

The training process is separated into two distinct phases. Initially, only the semantic and stochastic encoders are utilized, with the latent space DDIM remaining frozen. Upon convergence of the training, the two models are frozen, and the latent DDIM commences training. This model is tasked with learning the distribution of the semantic vector. The latent DDIM can be conditioned by a vector, similar to the stochastic encoder/decoder DDIM, thereby enabling the use of conditional generation.

6.4.3 Sampling

The sampling process is executed without the use of the semantic encoder, as this would necessitate the use of images as a condition. However, with the latent DDIM, it is possible to condition the sampling process with desired parameters. The semantic encoder can be employed to generate different stochastic variations of the input by passing it through the semantic encoder and using the resulting semantic vector as a condition for the stochastic encoder. This approach provides a flexible and efficient means of generating diverse stochastic variations of the input data.

Experiments and Results

In this chapter, various training and sampling strategies are explored and tested using proposed metrics. The primary focus is on evaluating the performance of different models and identifying optimal parameter settings based on these metrics.

7.1 Evaluation methodology

Evaluating the quality of the developed models was a complex task. Due to the limited size of the dataset, a conventional split between training and validation data was not feasible. The dataset, comprising 1279 images, is distributed over only 22 experiments. To address this challenge, a modified cross-validation approach was employed. Five experiments (numbers 3, 8, 11, 19, and 21) were chosen, and each model configuration was trained five times, excluding one of these experiments in each round. This selection was guided by the diversity in experimental settings, as some experiments shared similar parameters but varied in other, unmeasured factors. The chosen experiments were also distinct in the appearance of their resulting images. Full cross-validation was avoided due to the extensive time required; training a single model configuration took about 3 hours, resulting in a total of approximately 15 hours of computing time per model setting on an A100 GPU.

Post-training, each model configuration was used to generate samples under the same conditions as the experiment omitted during its training. The number of generated images matched the count in the respective dataset folder, ensuring a balanced comparison of distributions.

The initial metric selected for evaluation was the FID. This metric compares the distributions and their similarity. Given that FID does not inherently support conditional generation, it was adapted to meet the specific needs of this thesis. The FID was computed for each pair of folders containing either generated or real images, with the overall value derived from the average of these five computations.

To compare the electron spectra, an average image from each of the five experiments and an average image of samples with the corresponding conditional vector were created. The MSE metric was used for this comparison. This approach was beneficial as it emphasized models that accurately replicate both the overall intensity of the images and the position of the electron beam.

Another key quality of the sampling process is the ability of the models to simulate the stability of the electron beam. Experiments characterized by high variability in electron energy and quantity resulted in significant per-pixel variance in the images. Settings producing such data were termed unstable. The model's ability to replicate this instability was assessed by comparing the per-pixel variance between the original and sampled images. While the FID and MSE metrics indirectly measured this attribute, this specific focus on beam stability was taken as it is an

important factor for the application of the laser-plasma accelerator.

In summary, the chosen metrics provide a comprehensive framework for assessing the qualities of the generative process, offering a robust measure of the models' sampling accuracy.

7.2 DDIM performance

In the conducted analysis, the DDIMs were given primary focus due to their enhanced sampling speed, which facilitates easier optimization. The DDIM was trained with three variations in the number of noise steps and with three levels of physics information integrated into the loss function.

Two sampling parameters were assessed: the Classifier-free Guidance (CFG) scale and the number of denoising steps, referred to as sections, performed by the algorithm.

7.2.1 Physics-informed performance

Training of the model was carried out with different extents of physics information. In the initial phase, physics data was excluded from the loss function, with the emphasis placed solely on the comparison of noise in the images. This approach served as a baseline for subsequent performance evaluations.

For the first enhanced model, denoted as `phys_spec` in documentation and graphs, the incorporation of electron spectra MSE loss was executed as described in 6.3.1. The second variant, named `phys_spec_beam`, further included penalties for keeping non-zero values left of the beam-pointing pixel, as per the methodology described in 6.3.2. An improvement in performance across all metrics was anticipated for these physics-informed models, especially in the MSE of the spectra, due to the similarity of the loss function with the metric's computation.

As shown in Figure 7.1, the introduction of physics-based components in the model yielded varied results across different metrics. The `phys_spec` model was observed to have a lower FID in comparison to the model devoid of physics with median FID values 117.75 and 124.24, yet the `phys_spec_beam` model exhibited a decrease in performance compared to its less complex version with median FID of 120.65. The most physics-informed model also showed the highest variance in FID.

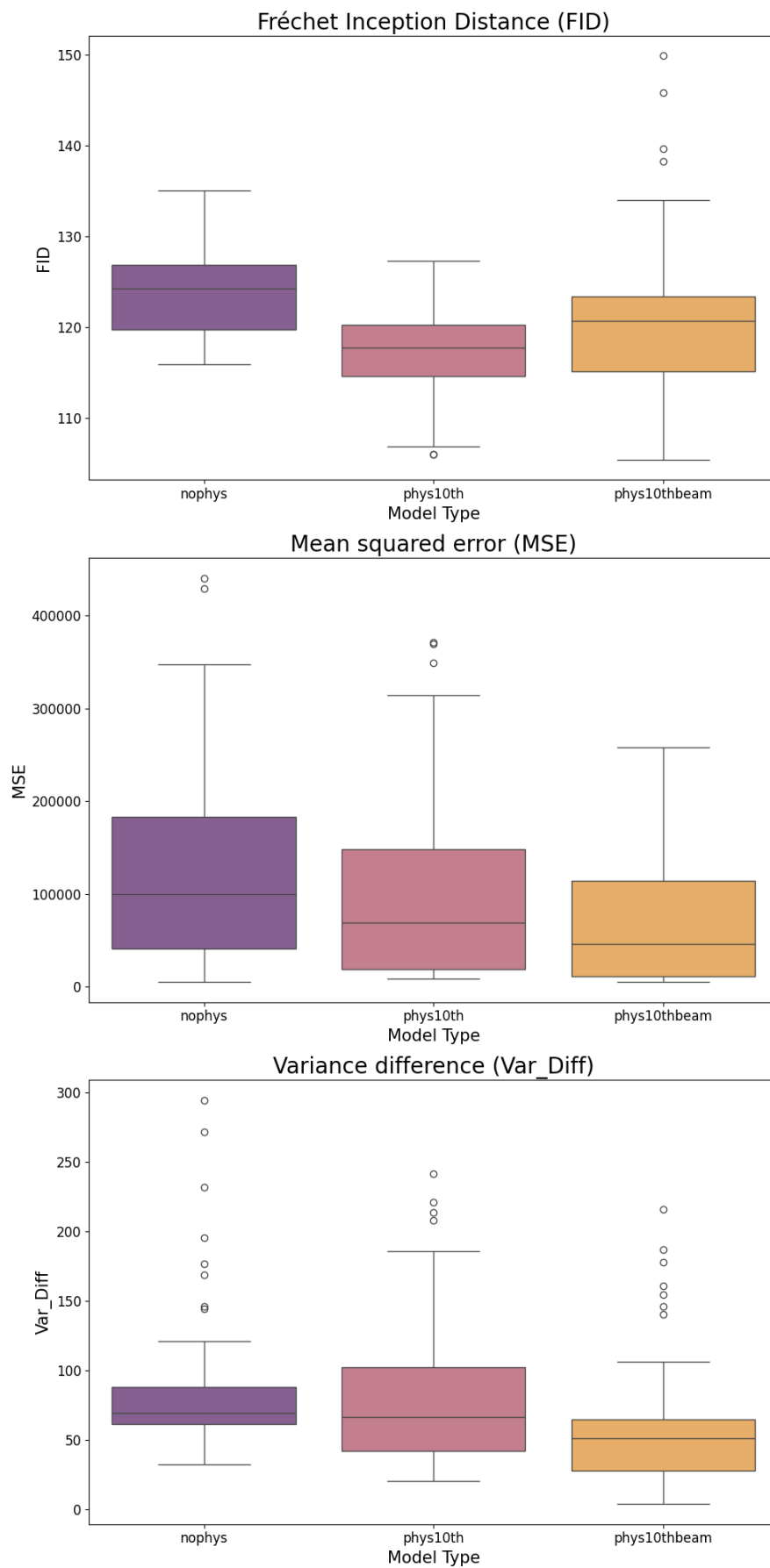
Observations from the data reveal that the models informed by physics exhibit greater variance in FID. This implies an increased sensitivity of such models to variations in sampling or training hyperparameters.

In MSE and variance difference the physics-informed models demonstrate superior performance, with the `phys_spec_beam` model outperforming the others with a median MSE of 46038.2 and variance difference of 51.43.

7.2.2 Analysis of noise steps

Initially, the model was trained with 1000 noise steps, which is commonly the standard for diffusion models. This number was reduced to decrease the time required for sampling, prior to the implementation of DDIM improvements. Additionally, it was hypothesized that a lower number of noise steps might be more effective, considering the relative simplicity of the visual data in this thesis. The dataset primarily consists of images with a dark background and a lighter area representing the electron beam, where the transition from dark to light is typically gradual. This simplicity contrasts with the complex shapes found in everyday scenes and objects, which are the usual subjects of other diffusion models.

According to the FID metric, as seen in 7.2, the results indicate that the two physics-informed models slightly outperform the `nophys` model in all noise settings, except at 1000 noise steps where the `phys_spec_beam` model shows poorer performance. The 700 noise steps setting exhibits



■ **Figure 7.1** Comparison of different model versions.

the least variability, while the 850 noise steps setting achieves the best performance with model type `phys_spec` at median FID 116.64 followed closely by the `phys_spec_beam` type at 116.97.

Concerning the MSE metric, the models trained with higher noise steps (850 and 1000) significantly outperform the model trained with 700 noise steps. According to this metric the model `phys_spec_beam` performed the best with a median MSE of 35822. On the other hand, the variance difference metric suggests that the model with the lowest noise steps setting most effectively replicates the stability of the electron beam. The model with the lowest median variance difference is the `phys_spec_beam` with 700 noise steps at 17.53.

These comprehensive findings are visualized in Figure 7.2. It is observed that the physics-informed models generally surpass the `nophys` model in most settings and metrics. Notable exceptions include the `phys_spec` model underperforming at 700 noise steps in the MSE metric and at 850 noise steps in the variance difference metric. In summary, a higher number of noise steps in training tends to benefit the first two metrics excluding variance difference. Therefore, if simulating beam stability is the primary objective, a lower number of training noise steps is recommended.

7.2.3 CFG statistics

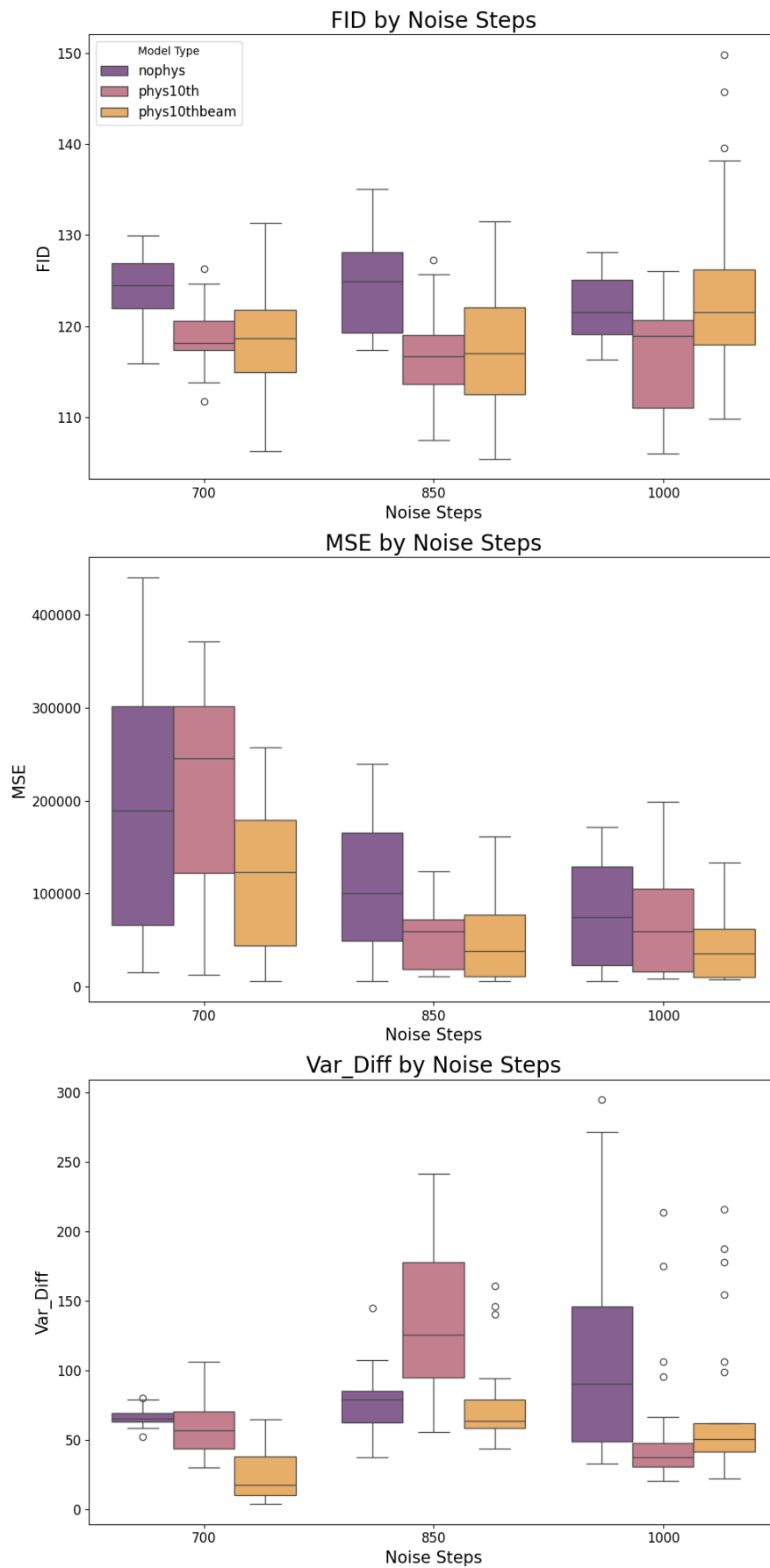
CFG modulates the influence of the conditional vector on sample generation by adjusting the weight of the conditional sample during its interpolation with an unconditional sample. Typically, a CFG value greater than 1 is employed to enhance the guidance provided by the settings vector. In this research, CFG values of 1, 3, 5, 6, and 7 were examined in detail.

Figure 7.3 illustrates that the impact of CFG on FID is minimal, with a CFG setting of 1 emerging as the most effective. The `phys_spec_beam` model type is identified as the best performer, though its performance appears to decline as CFG values increase. The superiority of a CFG value of 1 is further asserted by the MSE metric. According to this metric higher CFG values diminish the model’s ability to sample results accurately. The influence of CFG on the variance difference metric is relatively minor in terms of median values. However, an increase in CFG value leads to a significant rise in the variance of the variance difference metric.

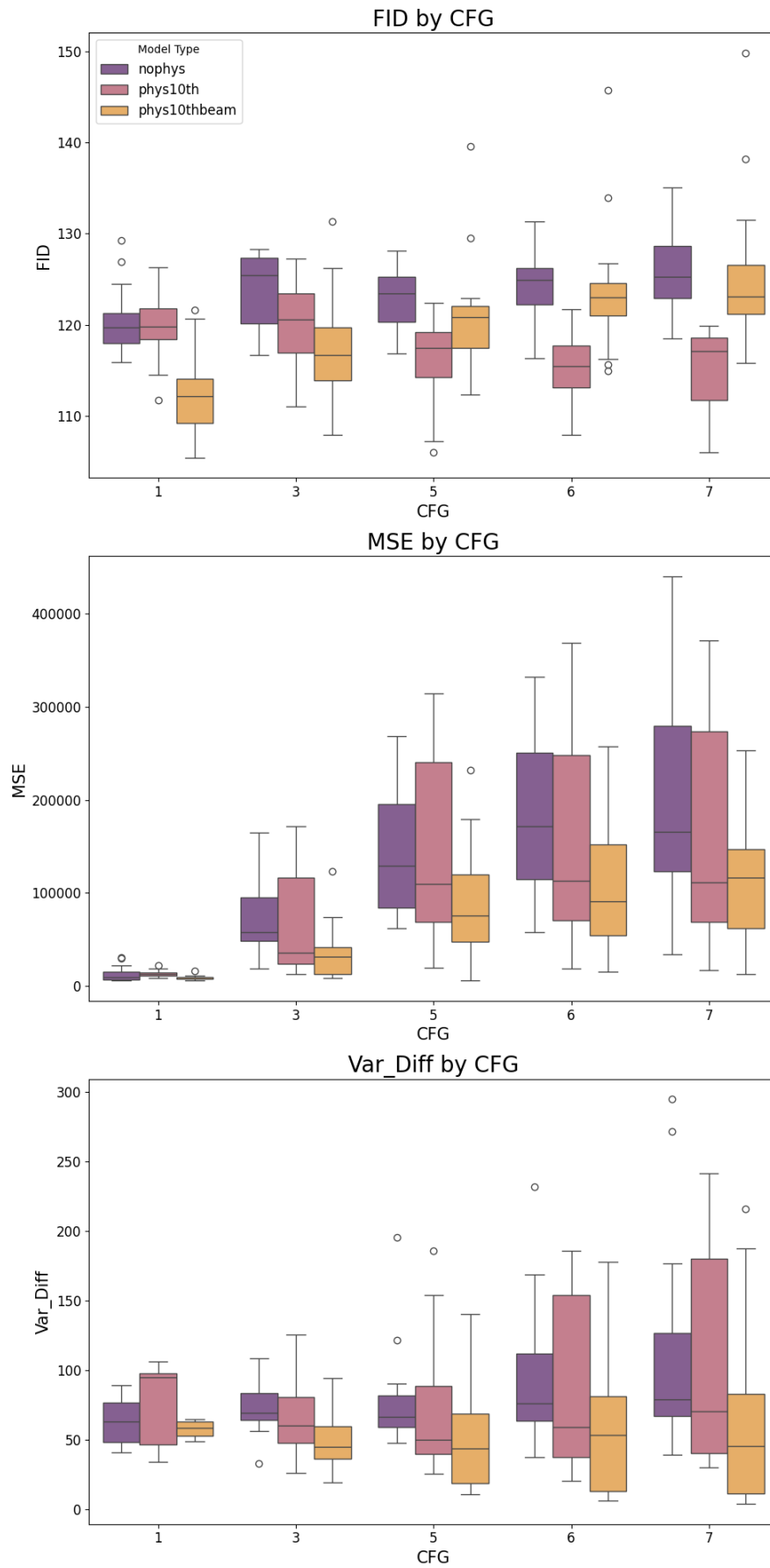
These overall findings reaffirm the superior performance of the physics-informed models, with the CFG 1 setting being the only scenario where the `phys_spec` model is outperformed by the `nophys` variant. Generally, lower CFG values yield better results, with the CFG 1 setting and the `phys_spec_beam` model emerging as the top performers in two out of the three metrics evaluated with median FID 112.14, MSE 9040 and variance difference 58.36. The best median variance difference is achieved by CFG 5 at 43.74.

7.2.4 Section count analysis

The “section counts” hyperparameter is unique to the DDIMs and plays a crucial role in the sampling process. This parameter signifies the number of denoising steps executed during sampling. While the model training incorporates 700, 850, or 1000 steps, the sampling phase doesn’t necessarily utilize all these steps. The term “section counts” arises from the approach where these steps can be evenly distributed across the training steps or specified as a list of values. The length of this list defines the number of sections, and the values indicate the desired number of noise steps within each section. This approach is particularly relevant to this thesis as the physics-informed models employed physics calculations only in the final 10% of the noising steps. For this analysis, section counts of 15, 25, and 45 were chosen, evenly sampling from the training noise steps. Additionally, configurations labeled “9+6” and “18+7” were employed. These settings divide the training noise steps into 10 sections, conducting 1 or 2 steps in the initial 9 sections and 6 or 7 steps in the last, totaling 15 and 25 new noise steps, respectively. These configurations were selected to evaluate whether more detailed sampling toward the end of the process is advantageous, especially for physics-informed models.



■ **Figure 7.2** Influence of different noise steps values on metrics.



■ **Figure 7.3** Influence of different CFG values on metrics.

The comprehensive outcomes of this examination are presented in Figure 7.4. In terms of the FID metric, the hypothesis that focusing on the final denoising steps would enhance performance is not supported. The results indicate that all three model types show a decline in performance when concentrating on the last steps of denoising. It's also observed that the `phys_spec` model outperforms the other models under all section count settings, with 45 denoising steps achieving the best performance, having a median FID of 114.72. Interestingly, minimal variance is seen between the 15 and 25 section counts. This finding suggests that 15 denoising steps may be preferable due to its faster sampling speed, a factor extensively discussed in the subsequent section.

In the context of the MSE metric, a similar trend is noted, with performance deteriorating when detailed denoising is applied towards the end. Contrary to the FID results, the 15 section count setting demonstrates the best median performance, with both physics-informed models yielding closely matched results at 21549 and 21399 median MSE.

For accurate assessment of beam stability, the “18+7” model emerges as the most effective, exhibiting a median variance difference of 43.31, closely followed by the “9+6” setting at 45.2. This metric is the only one showing improvement when focusing on the latter stages of the denoising process. However, this positive shift is observed across all three model types, including the physically uninformed one, suggesting that the advantage does not originate from the physics-based loss calculation.

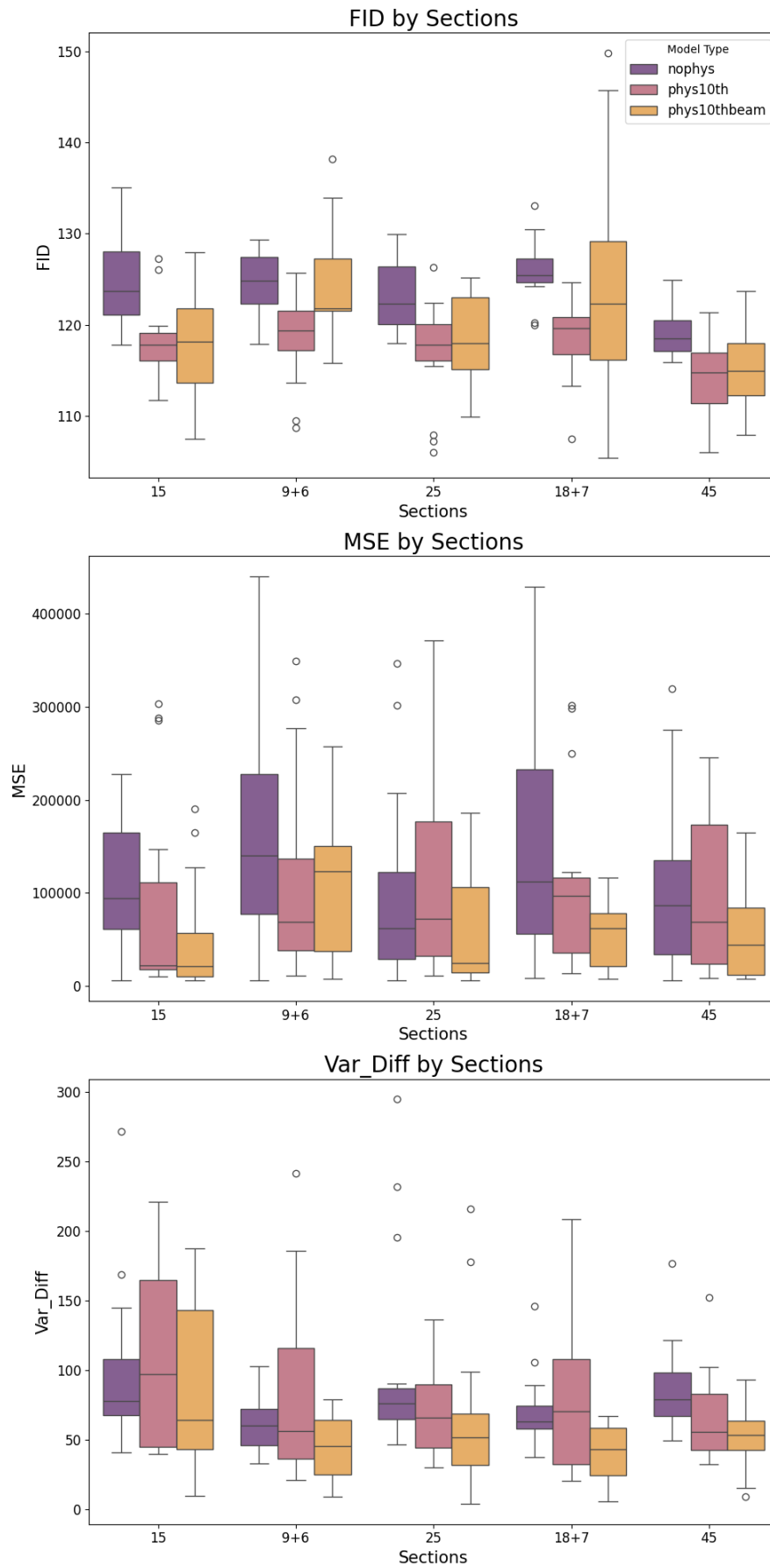
7.3 Sampling speed

Sampling speed is a critical factor, especially when models are intended for use in optimization tasks. In such scenarios, the model is frequently employed to generate new data based on specific setting vectors suggested by an optimization algorithm. As the optimization process involves numerous iterations to reach an optimal solution, the speed at which samples are generated directly impacts the overall duration needed for fine-tuning experimental parameters.

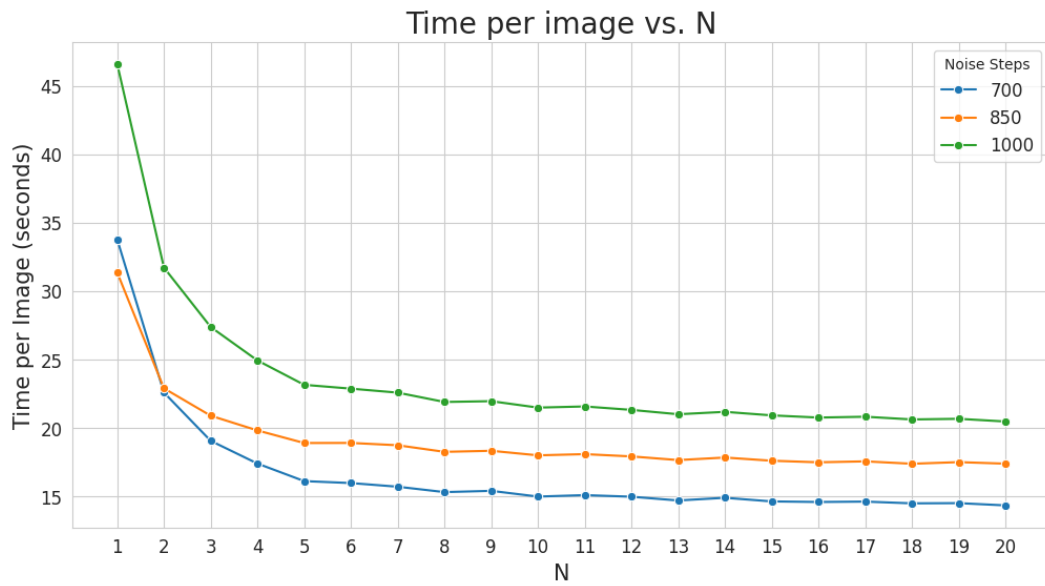
The traditional DDPM approach presents limitations in this context. The DDPM algorithm requires executing all training noise steps for each image to achieve a fully denoised output. Consequently, the image must be processed through the model for each noise step, making the number of training noise steps a key determinant of sampling speed. Sampling allows for batch processing of multiple images that can decrease the time per image but substantially increase VRAM usage. The largest tested batch size of 20 required approximately 28GB of VRAM on an A100 graphics card. Figure 7.5 demonstrates a time comparison for generating a single image using different batch sizes and training noise steps with the DDPM. The fastest time recorded per image with DDPM is 14.33 seconds, using the 700 noise steps setting and a batch size of 20. However, this duration is still relatively long and constraints training to use as few noise steps as possible.

In contrast, the DDIM methodology is far more efficient for rapid image generation. The DDIM sampling process leverages the model only during a new set of noise steps, sampled from the original training steps. The comparative analysis of the time required per image using the DDIM is depicted in Figure 7.6. For this analysis, the same number of sections used in the qualitative evaluation of the models was employed. Notably, the DDIM achieves impressive speeds of around 0.32 seconds per image in the 15-section setting at a batch size of 8. At this batch size, the 25 and 45-section settings yield speeds of 0.55 and 0.98 seconds per image, respectively. The VRAM requirement for this batch size is a more manageable 11.7GB.

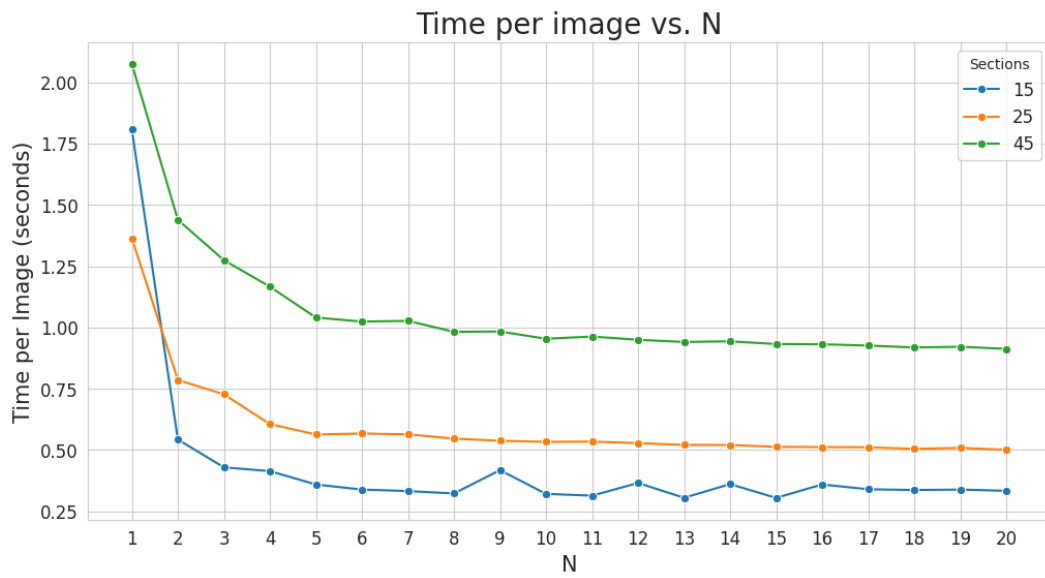
The comparative analysis between DDPM and DDIM highlights the significant advantage of DDIM in terms of speed and efficiency. This makes the DDIM approach particularly suitable for applications where rapid sampling is important, such as in iterative optimization tasks where model responsiveness can significantly influence overall performance and efficiency.



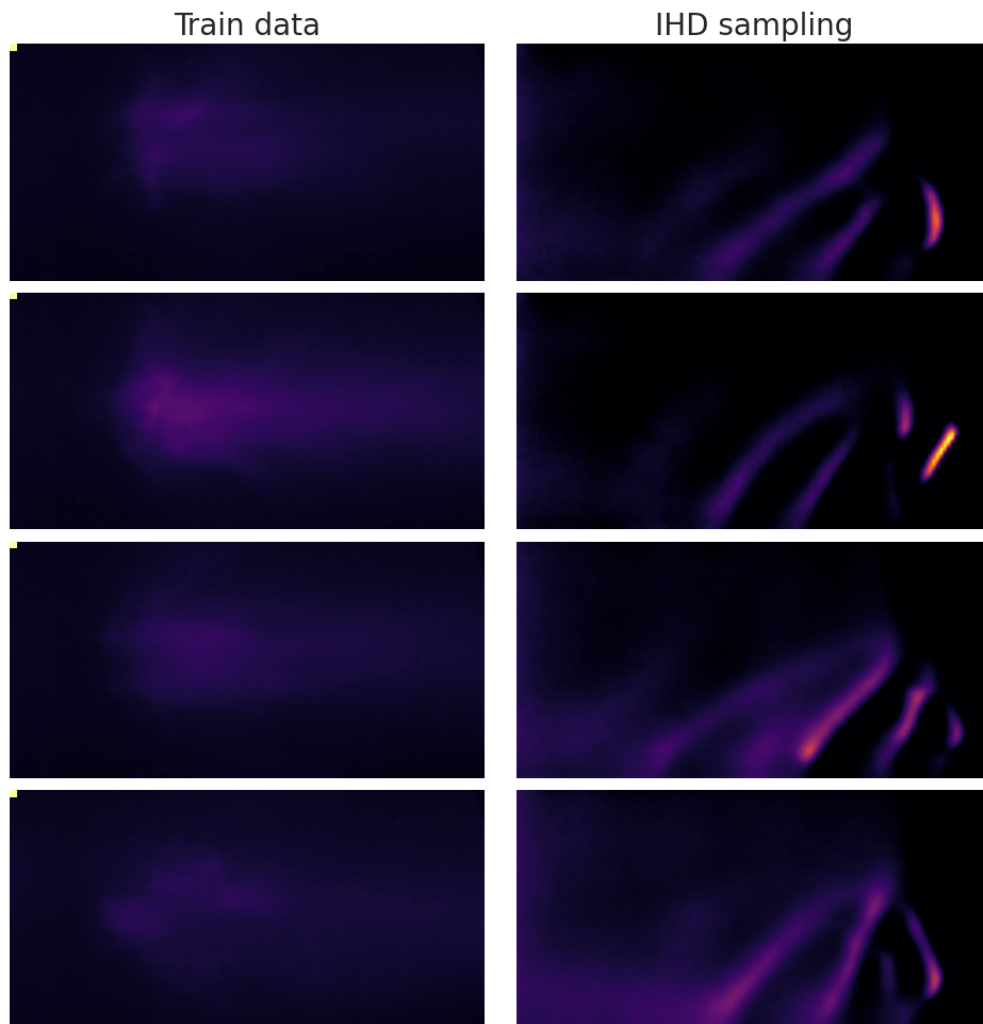
■ **Figure 7.4** Influence of different sections values on metrics.



■ **Figure 7.5** Influence of noise steps and sampling batch size on the generation speed using the DDPM algorithm.



■ **Figure 7.6** Impact of sections and sampling batch size on generation speed using the DDIM algorithm.



■ **Figure 7.7** Showcase of performance of the IHD model. The left column shows images from the training dataset while the right shows generations given the same settings vector.

7.4 Inverse heat dissipation (IHD)

The blurring of images as a substitution for progressive Gaussian noising was also explored. The different noising process accepted different parameters of the noising schedule. These were the blurring schedule parameters σ_{min} and σ_{max} and the δ parameter of the sampling process. These parameters were set according to the official implementation's suggestions to the original implementation [47]. The results of this setting were unsatisfactory. Several attempts were made to improve sampling quality either by changing sampling parameters or making adjustments to the training loop. The results of this approach still stayed far behind what was already achieved by standard Gaussian noising and therefore the IHD approach was abandoned as not promising. In Figure 7.7 the sampling process of the model can be assessed when compared to the images with the same conditional vector from the training dataset.

7.5 DiffAE results

The experimentation with the diffusion autoencoder yielded results that were below expectations. The scope of training was limited to the initial stage, involving the semantic encoder and the stochastic encoder/decoder. The efficacy of this training phase was assessed by generating stochastic variations of images. This process entailed feeding an image through the semantic encoder, then using the resultant semantic vector as a conditional input for the stochastic decoder’s sampling algorithm. The outcomes of this process are illustrated in Figure 7.8. In some instances, as evident in variation 2 of the figure, the model was able to reconstruct an image bearing a faint resemblance to the original input. However, the model’s performance was predominantly erratic, with the generated variations often bearing little similarity to the input image, sometimes resembling entirely different experiments with dissimilar settings, or, in most cases, producing mere noise.

A range of hyperparameter adjustments were tested, along with a reimplementaion of the conditional vector processing, in an effort to refine the model’s performance. Despite these efforts, there was negligible improvement in the quality of the output. This led to the conclusion that, while training a DDIM on the available dataset is feasible, the volume of available data is likely insufficient for the successful training of an autoencoder model.

Given the insufficient quality of outcomes of the initial training phase, the subsequent development, training, and evaluation of the latent space DDIM model were not pursued. The challenges encountered in the earlier stages indicated that further efforts in this direction are reasonable after collecting data from more real-world experiments.

7.6 Evaluation of best configurations

In our comprehensive analysis, a total of 225 models were evaluated, encompassing various levels of physics integration, noise steps, CFG scales, and section counts. Two models emerged as leading candidates based on their performance across multiple metrics. The selection process involved a systematic evaluation, wherein each model’s average rank was calculated by considering its performance across all metrics.

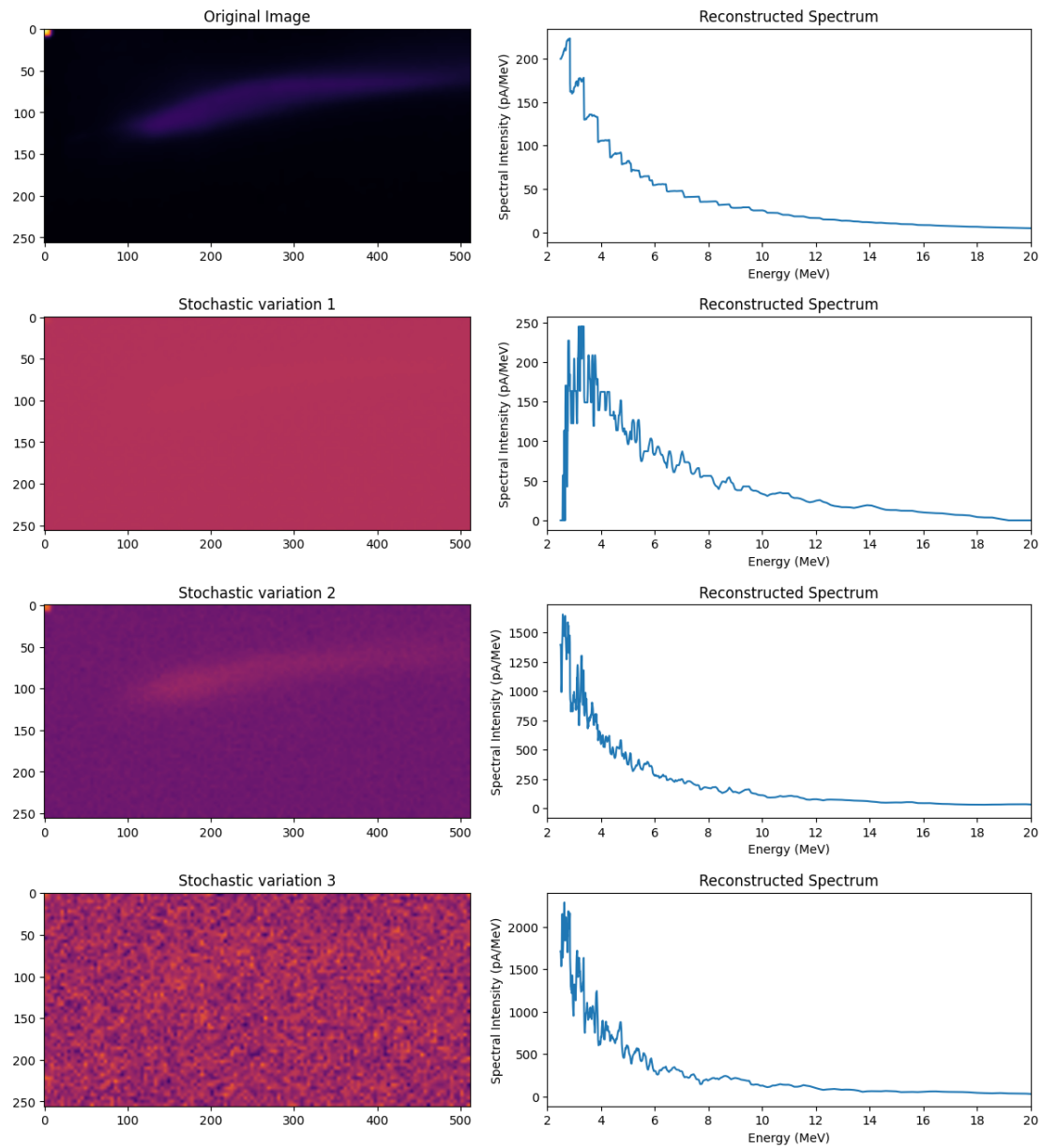
The first selected model shown in 7.1 distinguished itself as the most proficient, not only in the average rank across all metrics but also specifically in FID and the combination of FID and MSE metrics. However, its performance in the variance difference metric was less impressive, ranking 100th. To address this, a second model was identified, placing a greater emphasis on the variance difference metric. The ranking process was adjusted by doubling the weight of the variance difference metric, leading to the selection of this second model as another top candidate. Detailed settings and performance statistics of both models are presented in 7.1.

Model Type	Steps	CFG	Sections	FID(rank)	MSE(rank)	Var_diff(rank)
phys_spec_beam	850	1	18+7	105.38 (1)	7100 (9)	59.84 (100)
phys_spec	1000	3	18+7	116.16 (50)	16076 (48)	27.02 (23)

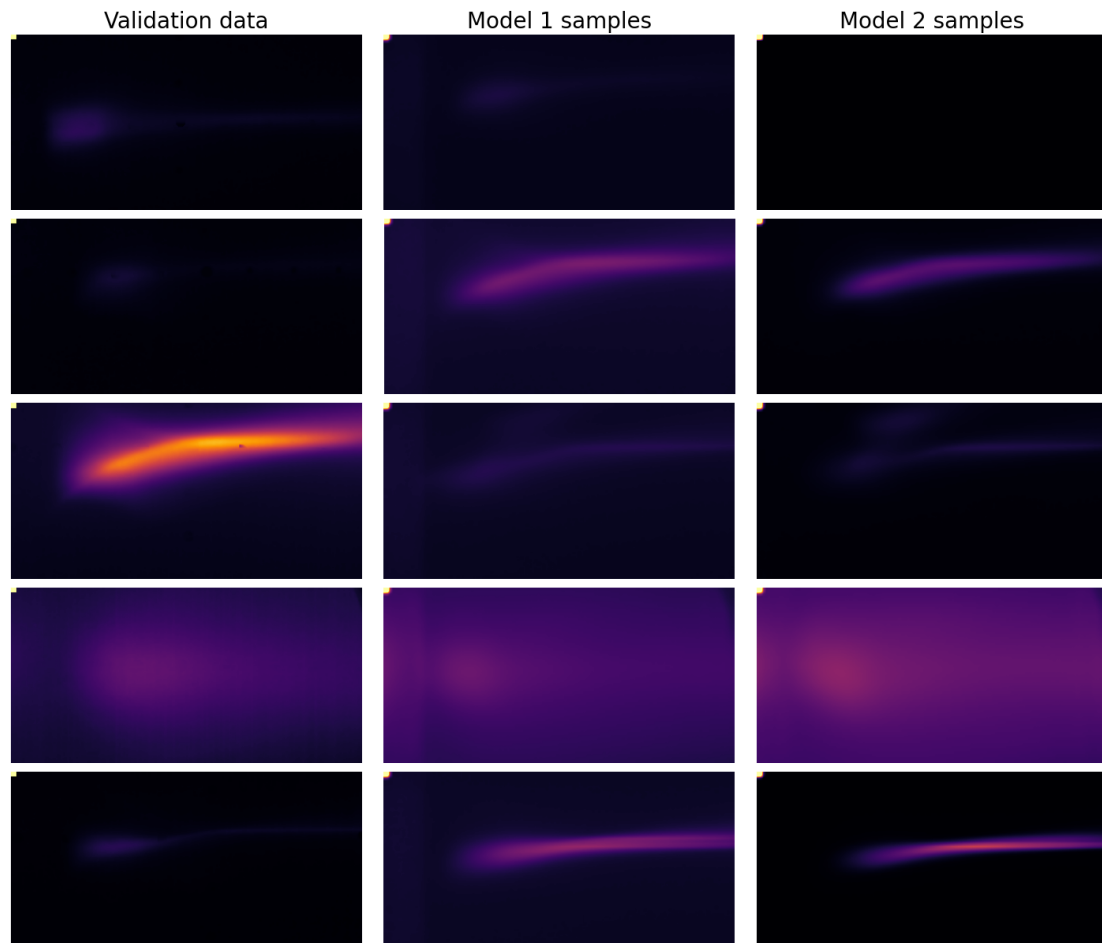
■ **Table 7.1** Best models according to the selected metrics. The numbers in brackets indicate the rank of the model in the corresponding metric.

The effectiveness of these models can be further understood by analyzing Figure 7.9 by comparing average images of validation data. Both models exhibit comparable performance, with the model focused on variance difference demonstrating a closer approximation to the correct background values.

It is important to note that the comparison of average generated images does not factor in variance. Therefore, to evaluate the models more rigorously, the most variable validation experiment was selected, and the models were tasked with sampling this specific data. The



■ **Figure 7.8** An input image with its electron beam spectrum (top row) and its 3 different stochastic variations.



■ **Figure 7.9** Comparison of the performance of the two best models. The left column shows an average training image per validation experiment. “Model 1” is the `phys_spec_beam` model expected to perform best across all metrics and “Model 2” is the best model when variance difference weight is doubled. The intensity of the images is doubled for ease of viewing.

outcomes of this sampling are depicted in Figure 7.10. While both models show similar levels of performance, the model emphasizing variance difference appears to have a slight edge in replicating the variability between samples. The samples from the first model show variations primarily in overall intensity, whereas the second model demonstrates more variability in the position of the electrons, aligning more closely with the characteristics of the validation data. This nuanced difference in performance highlights the importance of considering specific metric emphases in model selection, particularly in applications where certain aspects, such as electron beam stability, play a critical role.

7.6.1 Sampling of random data points

The use of the models is primarily to sample from unknown places within the distribution from which the training dataset originates. These results are used as outputs of a simulation of the physics experiments. It is difficult to evaluate how good these generations are since there is no ground truth data to compare them to.

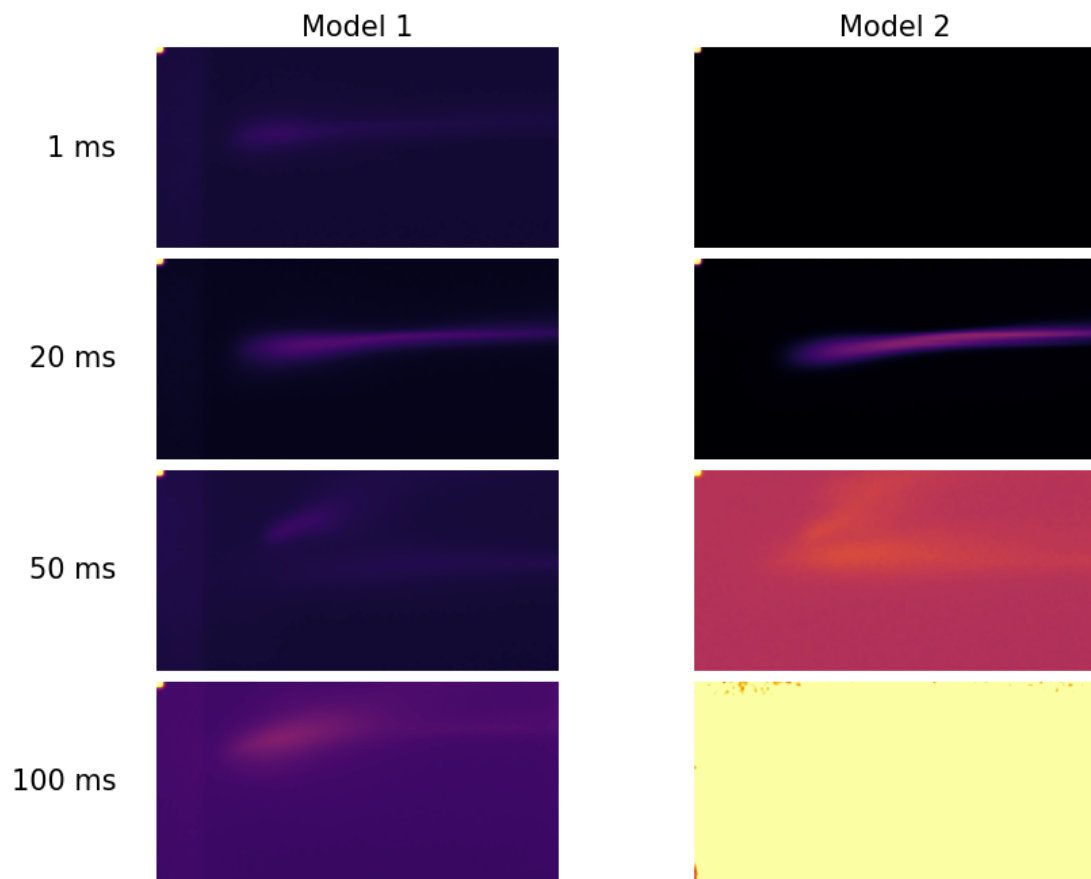
The only experimental parameter with a known influence is the acquisition time. Generally speaking, two experiments with the same energy and pressure settings that differ in acquisition time should differ in average intensity. Lower acquisition time leads to lower intensity values and vice versa. We can test the two candidate models based on these assumptions. Figure 7.11 shows average images from both models when energy is fixed to 15 mJ, gas pressure to 15 bar and acquisition time is either 1, 20 or 100 milliseconds. It can be seen that Model 2 shows a better understanding of the acquisition time parameter by changing the intensity of the image without changing the shape of the beam spectra.

The performance of the two candidates for the best model was discussed with the experts at ELI Beamlines. The overall consensus was that Model 2 is better at sampling with parameters approximately within the range of performed experiments and is valuable for simulations when laser is not available as a substitution. Model 1 was praised for its ability to meaningfully predict the behavior of the system when conditioned by parameters with values well outside the normal operating range. This model was suggested to be used when assessing safety measures for experiments with higher laser energy to help approximate the expected outcomes of new experiments.

To provide a more comprehensive understanding and visual representation of the model's capabilities in generating samples, various examples have been included in the Appendix at the end of this thesis. These examples serve to illustrate the types of outputs the models are capable of producing, offering insights into their potential utility and limitations in simulating aspects of the physics experiments under consideration. While these examples cannot substitute for a comparison with actual ground truth data, they do provide valuable evidence of the model's performance and the feasibility of using such models for simulating complex physical phenomena.



■ **Figure 7.10** Comparison of sampling variance of the two best models. The left column shows random images from the validation folder. “Model 1” stands for the overall best `phys_spec_beam` model setting and “Model 2” is the one more focused on variance difference.



■ **Figure 7.11** Comparison of model performance when changing the acquisition time parameter. The different values can be read on the left and models on the top. Model 1 is the generally best model, Model 2 is the variance-focused one. Each image is an average of 16 generated samples.

..... Chapter 8

Future work

The research presented in this thesis represents a significant contribution to the intersection of laser-plasma particle accelerator optimization and machine learning. However, the scope of this topic extends far beyond the bounds of this work, encompassing many possibilities in both experimental physics and the application of advanced machine-learning techniques. This chapter aims to outline potential avenues for future research in machine-learning-driven simulations and optimizations of physical experiments.

8.1 Modern diffusion methods

The field of diffusion models is rapidly evolving, with text-to-image generators leading the innovation charge. Newer iterations of these models are released periodically, each surpassing its predecessors in capabilities. These advancements are often the result of collaborative efforts among numerous machine learning experts. Recent developments [65, 5, 38] have introduced modifications in architecture compared to those discussed in this thesis. A prevailing trend is the shift away from performing diffusion in the original image space. Instead, a variational autoencoder (or a similar model) is first employed to create a compressed latent space. The diffusion process then takes place in this lower-dimensional space, significantly reducing computational demands while simultaneously enhancing the quality and fidelity of the generated data.

8.2 Autoencoder architecture

The autoencoder architecture, though incomplete in this thesis, holds substantial promise. Its capacity to differentiate between stochastic and semantic aspects of the acceleration process is pivotal for a deeper understanding of the underlying physics. The current limitations of the proposed model, primarily due to the size of the training dataset, can potentially be overcome through expanded collaboration with facilities like Eli-Beamlines. Enlarging the dataset would enable more robust training, potentially unlocking the full capabilities of the autoencoder approach.

8.3 Expanding physics research

The integration of data from multiple Alpha campaign runs presents its own set of challenges. The outcomes of these experiments are influenced by numerous variables, not all of which are consistently measured or documented. This leads to disparities between datasets where seemingly identical settings yield different results. To enable data merging, various constants,

such as those used for calculating electron position and energy, need reevaluation. Adjustments in the data preprocessing pipeline, like accommodating different camera positions that affect pixel-to-millimeter and pixel-to-radian conversions, are essential for successful integration.

Moreover, the architectures proposed in this thesis have the potential to be adapted for analyzing a range of experiments involving electron acceleration. With some modifications, they could even be applied to diverse image-producing experiments. Smaller-scale projects, such as accelerated particle detection and real-time particle energy estimation, could also be pursued. These tasks, while less complex, could significantly streamline the workflow of scientists and contribute to advancements in the field.

The synergy of machine learning and experimental physics, particularly in the realm of laser-plasma particle acceleration, offers many research opportunities. Future endeavors in this domain have the potential not only to advance our understanding of physical phenomena but also to propel the development of innovative machine-learning methodologies.

Conclusion

The main objective of this thesis was to conceptualize, design, and successfully train a generative model that functions as an optimization tool for laser-plasma particle accelerators. The core concept revolved around constructing a diffusion-based generative model capable of synthesizing images from a specified vector of input parameters. This endeavor aimed to bridge the gap between theoretical models and practical applications in particle accelerator optimization.

The path towards this goal required extensive data preprocessing. A comprehensive pipeline was developed to transform the raw experimental data into a format suitable for effective model training. This transformation process was critical in ensuring the quality and applicability of the training dataset.

A series of advanced Diffusion model architectures were explored, including the DDPM, DDIM and DiffAE models. Complementing these architectures, two data noising techniques were employed: traditional Gaussian noising and blurring via Inverse heat dissipation. Among these varied methodologies, the DDIM architecture, combined with Gaussian noising, demonstrated superior performance, particularly in terms of image generation speed - a critical factor in real-time applications.

In pursuit of the optimal model configuration, extensive hyperparameter tuning was conducted, balancing the model's performance across multiple metrics. The evaluation metrics were carefully chosen to reflect different aspects of model performance: FID for overall image quality, Variance difference for assessing the simulation of beam stability, and MSE of electron spectra for accurate simulation of beam charge.


Two hyperparameter sets emerged as most effective: one prioritizing a balanced performance across all metrics, and the other placing additional emphasis on beam stability. Both configurations attained impressive generation speeds of 1.8 images per second, underscoring their suitability for real-time optimization tasks.

Furthermore, this thesis shows the demonstrable advantage of physics-informed machine learning models over their uninformed counterparts. While the theory that sampling more steps toward the end of the backward diffusion process in physics-informed models is not backed by statistical evidence, both of the proposed best models are physics-informed and use this sampling strategy.

The developed model will be used for optimization that will output meaningful sets of parameters to be used during the next iteration of laser experiments. The data gathered from these experiments can then be used for re-training or finetuning the model, running the optimization again and proposing new parameters. This process may be repeated to help achieve an optimized electron acceleration experiment. Upon receiving more data a Diffusion Autoencoder model proposed in this thesis can be trained and used to better understand the intricate interactions in the experiments by separating the simulation process into its stochastic and semantic elements. These individual parts can be further analyzed to hopefully achieve a near-complete understanding

of the electron acceleration task.

The field of laser-plasma particle acceleration, particularly its optimization and simulation via machine learning, leaves many avenues for research and innovation. The methodologies and findings presented in this thesis contribute to this field and also pave the way for novel approaches and future research directions.

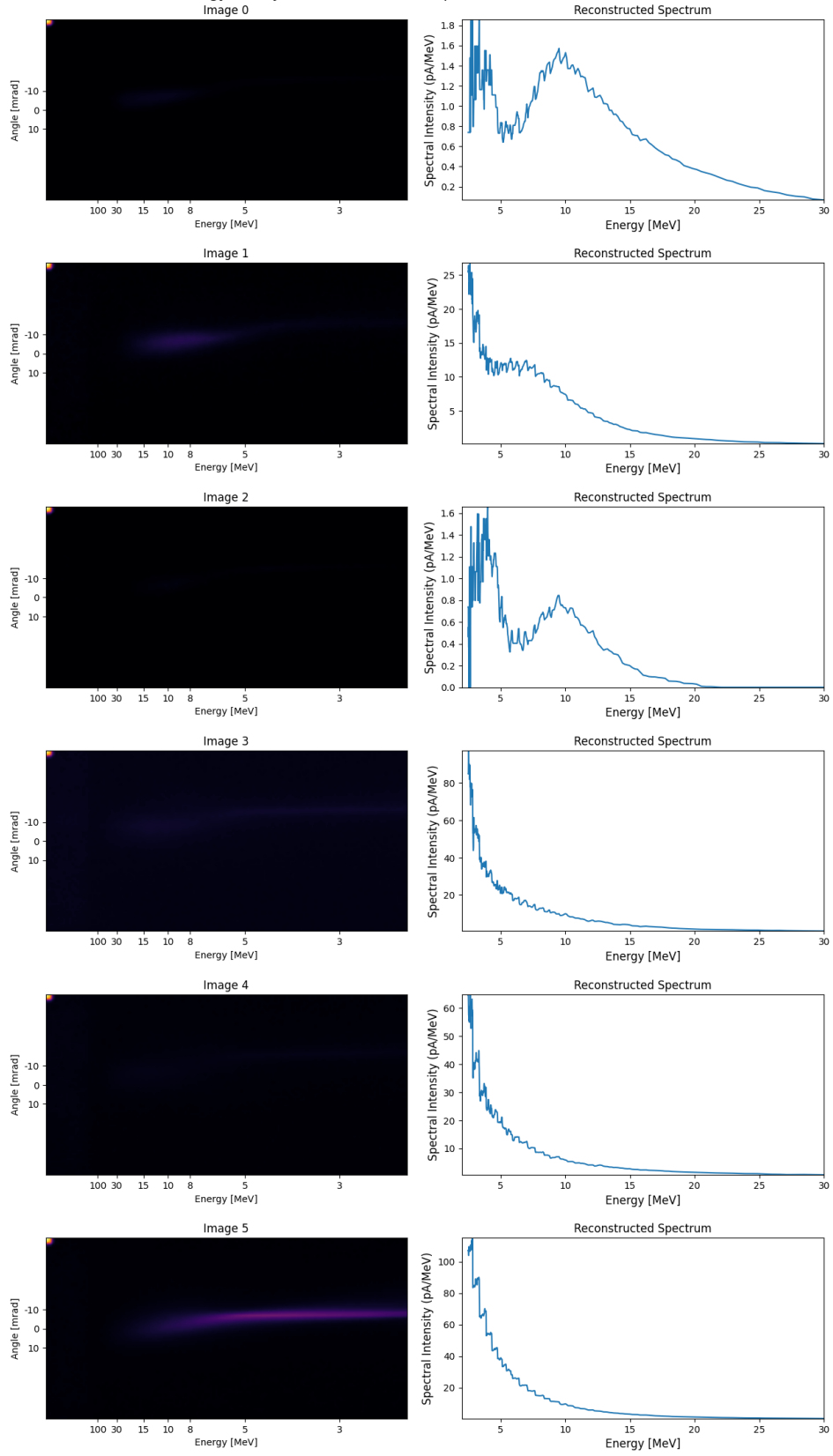


Appendix A

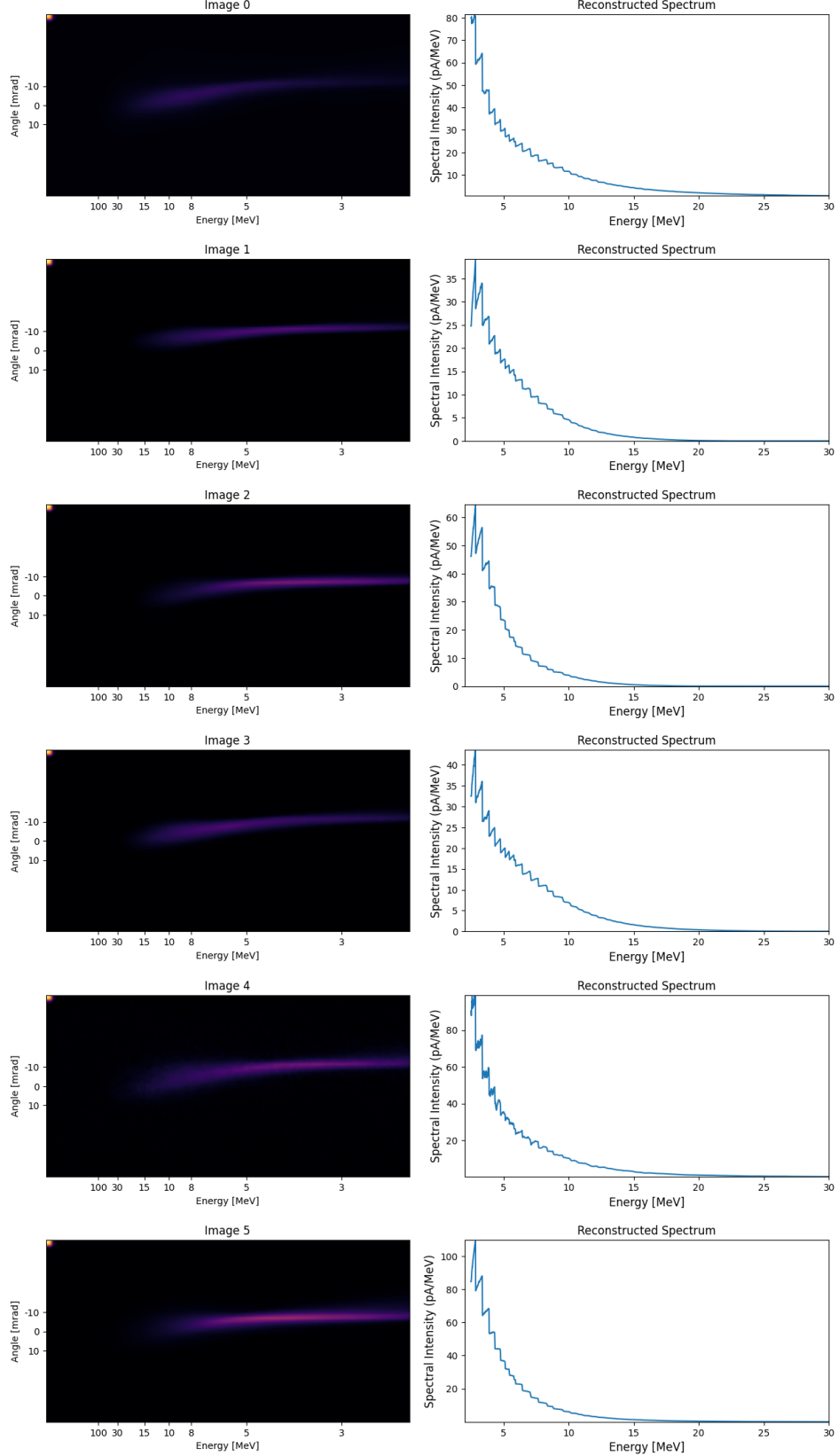
Examples of model sampling performance

Here some examples of generations by the two best models are shown. The parameters together with the ID of the model are situated at the top of the images. Each sample is listed together with its calculated electron beam spectrum.

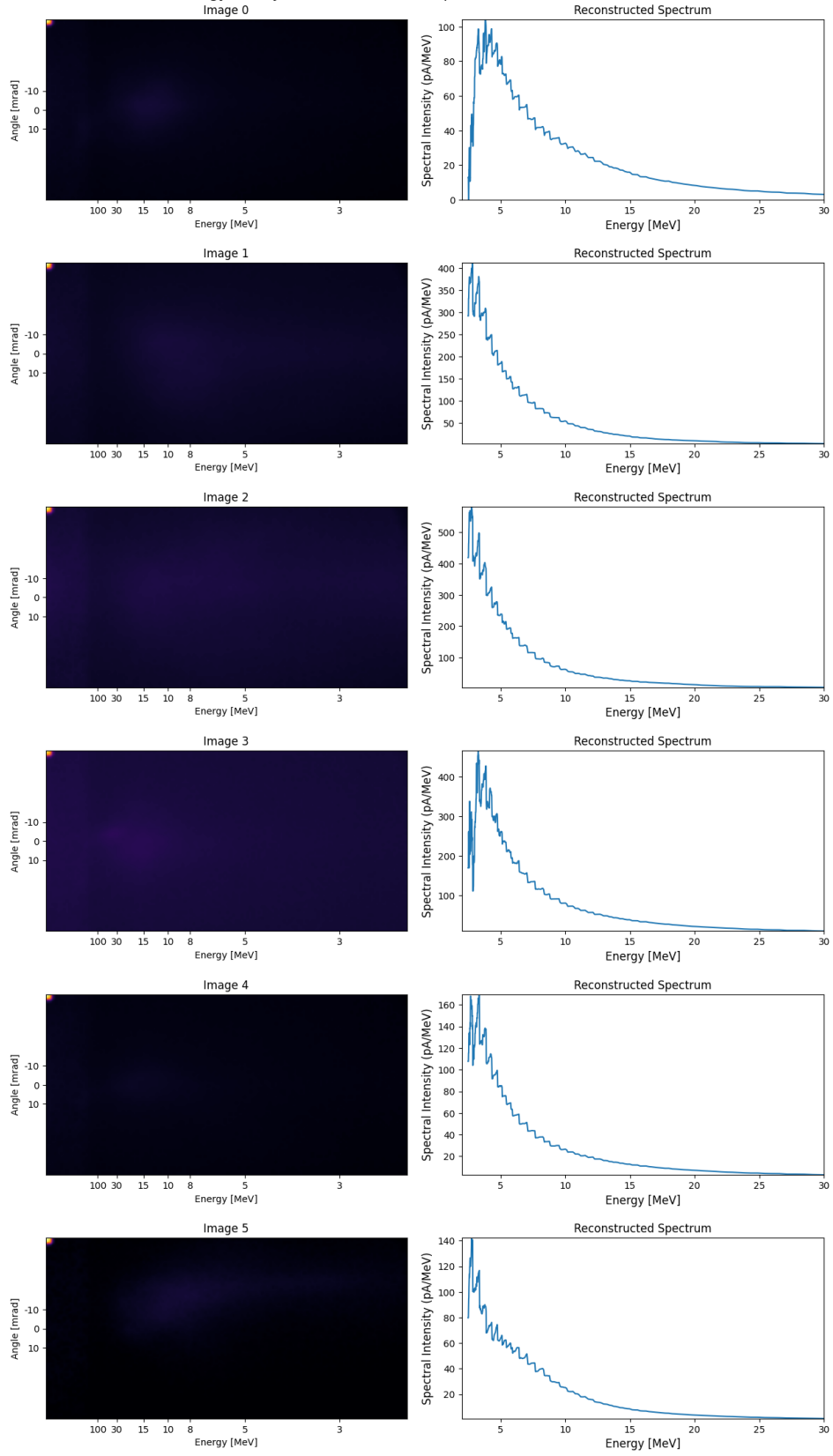
Energy: 15 mj, Pressure: 15 bar, Acquisition time: 25 ms, Model: 1



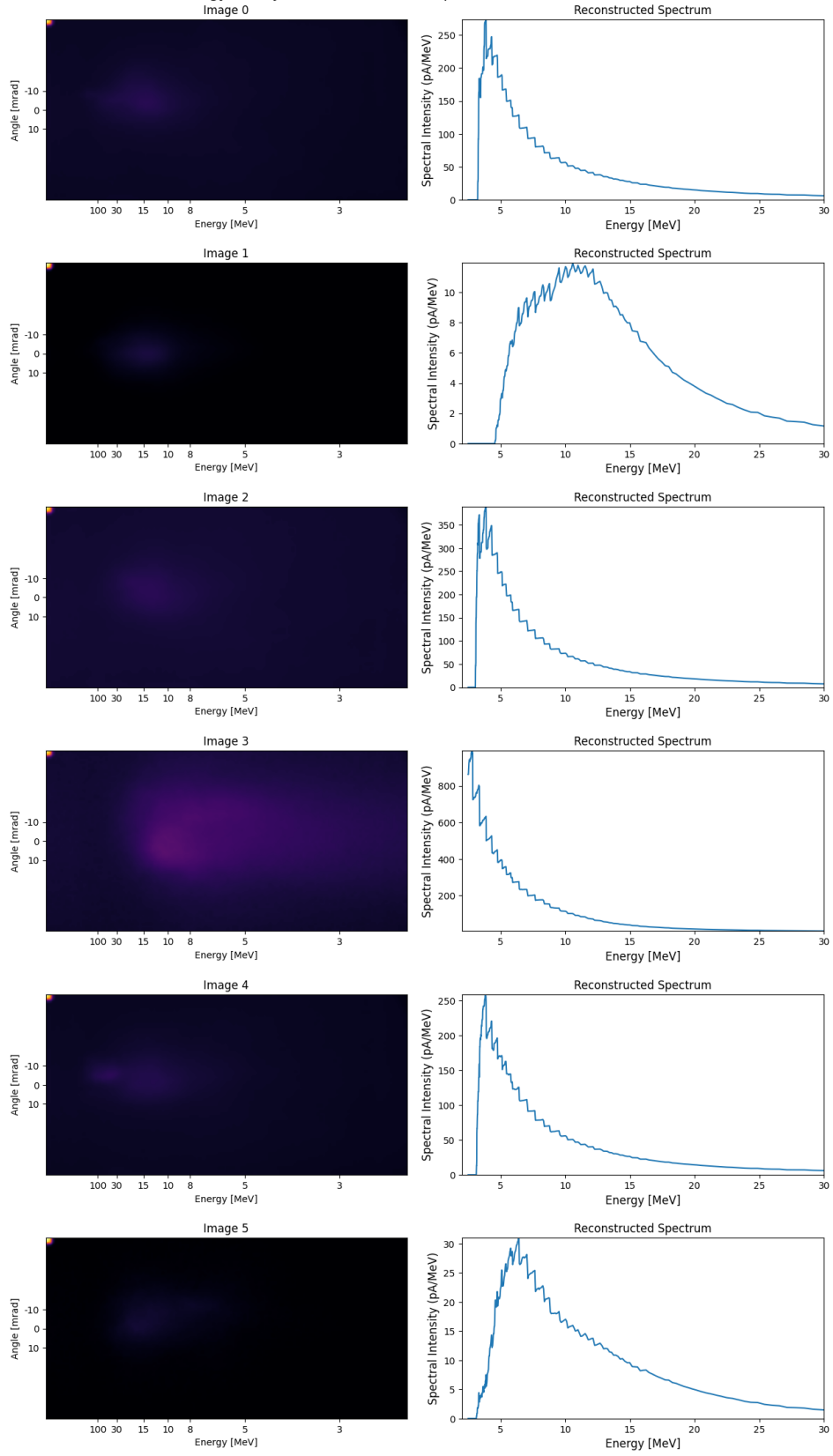
Energy: 15 mj, Pressure: 15 bar, Acquisition time: 25 ms, Model: 2



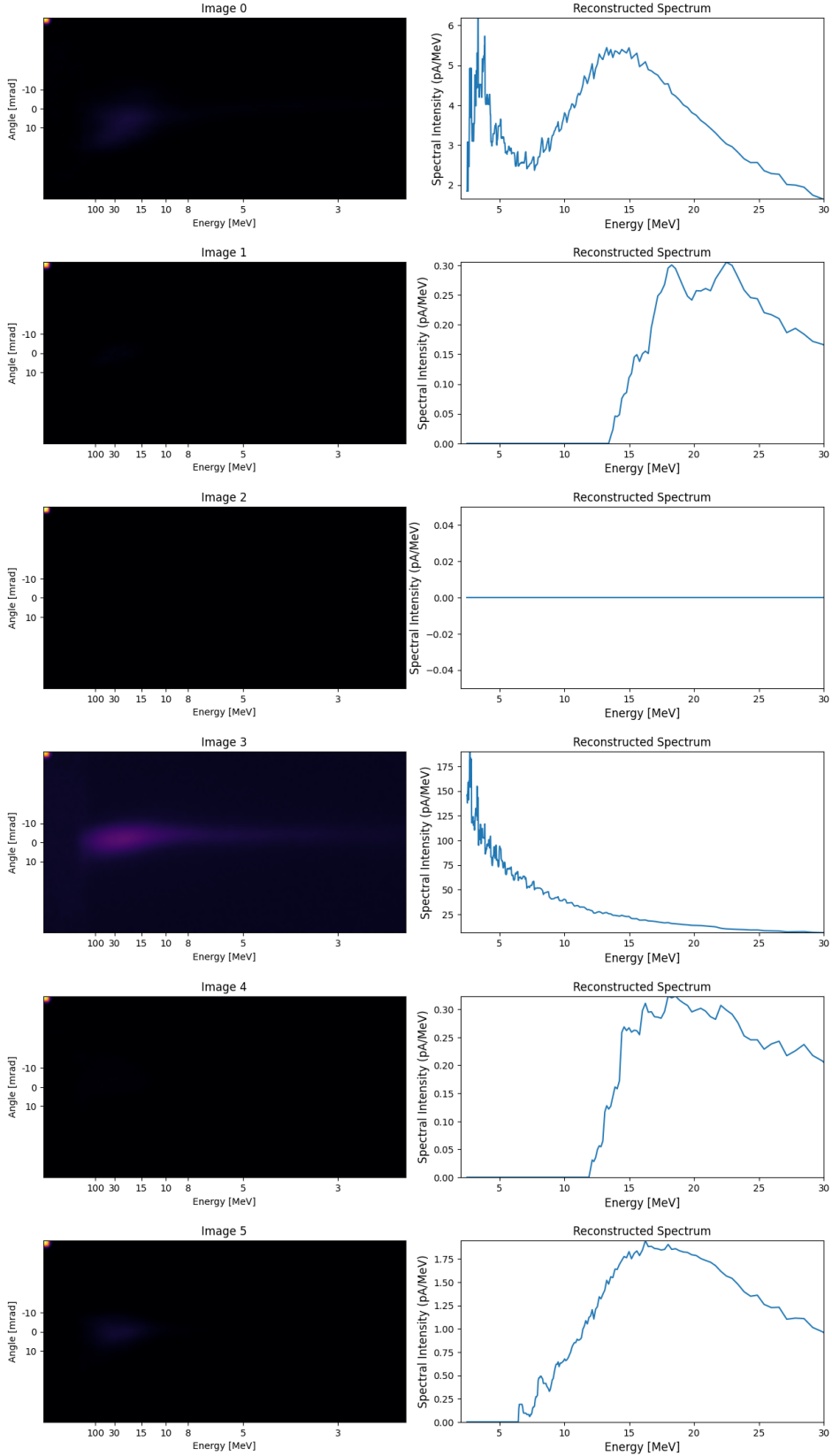
Energy: 25 mj, Pressure: 25 bar, Acquisition time: 15 ms, Model: 1



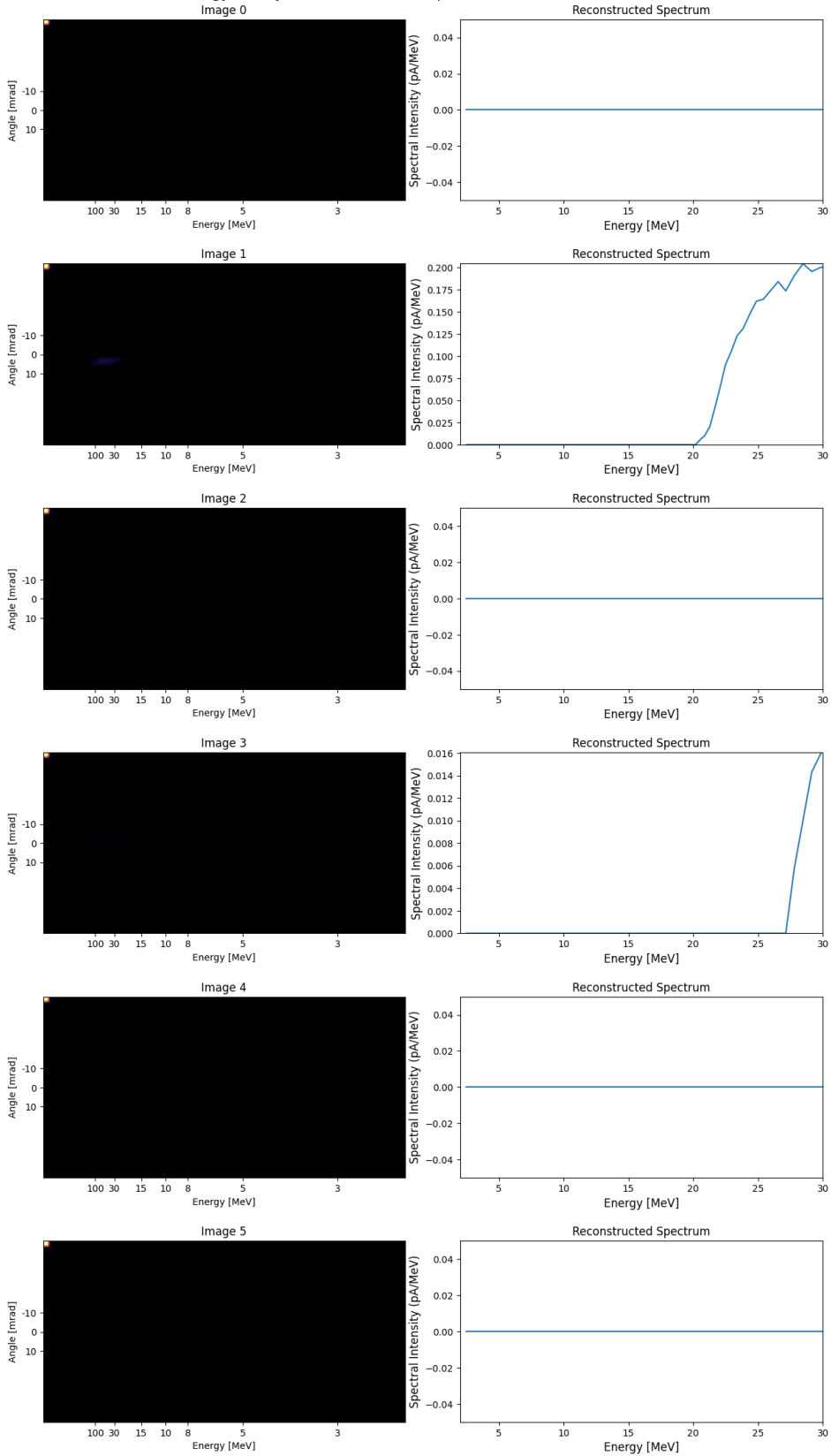
Energy: 25 mj, Pressure: 25 bar, Acquisition time: 15 ms, Model: 2



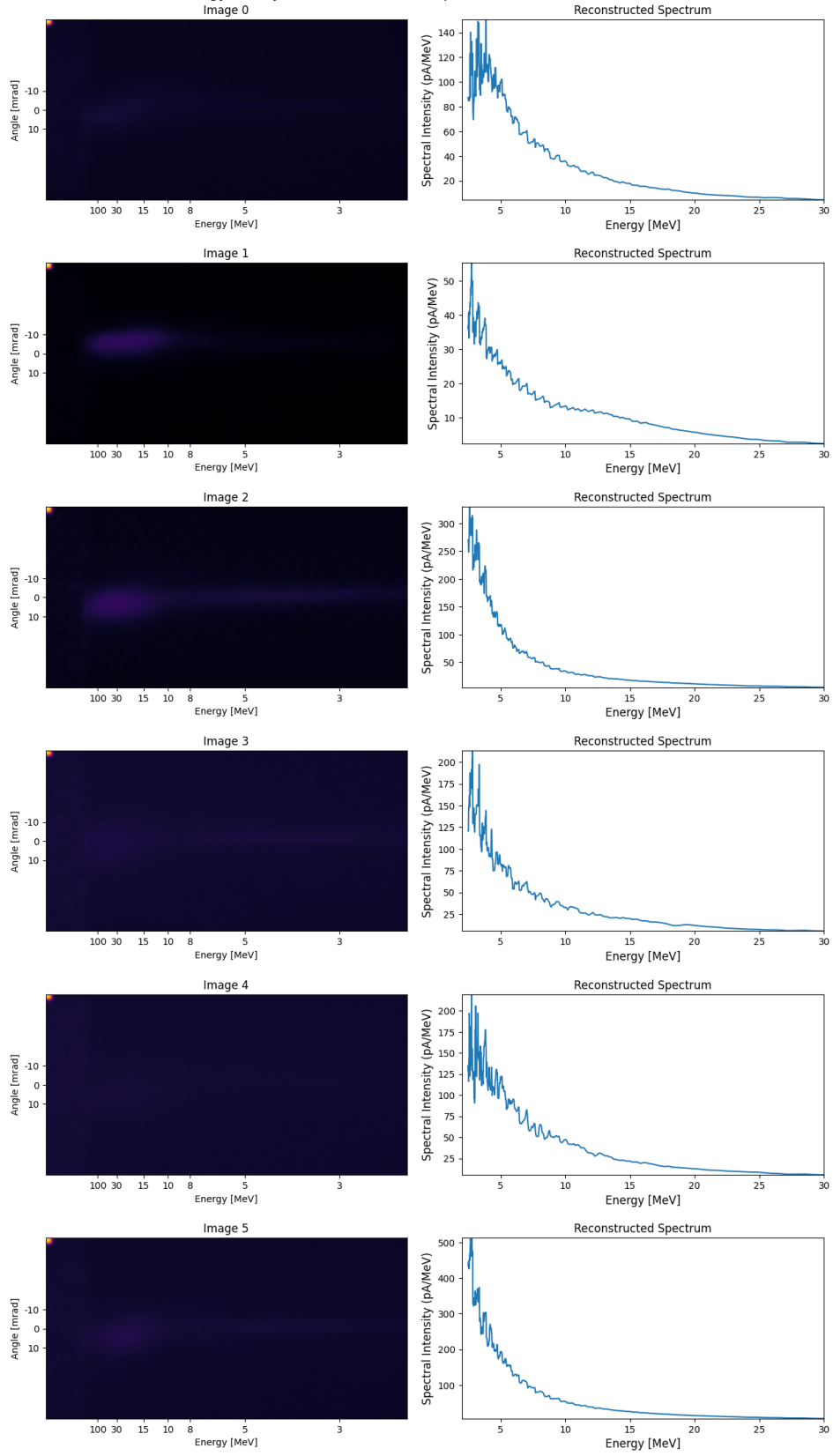
Energy: 32 mJ, Pressure: 4 bar, Acquisition time: 15 ms, Model: 1



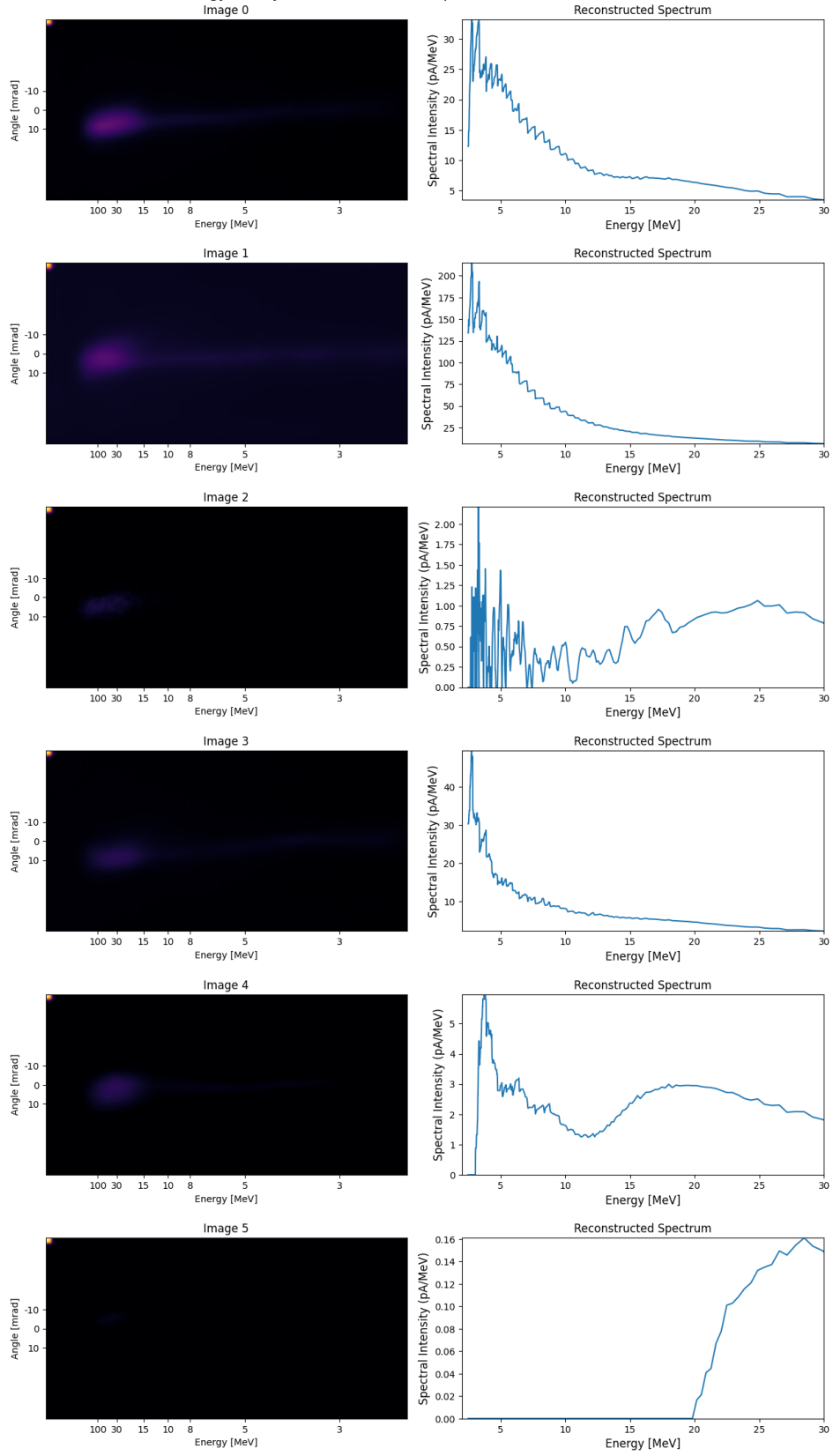
Energy: 32 mJ, Pressure: 4 bar, Acquisition time: 15 ms, Model: 2



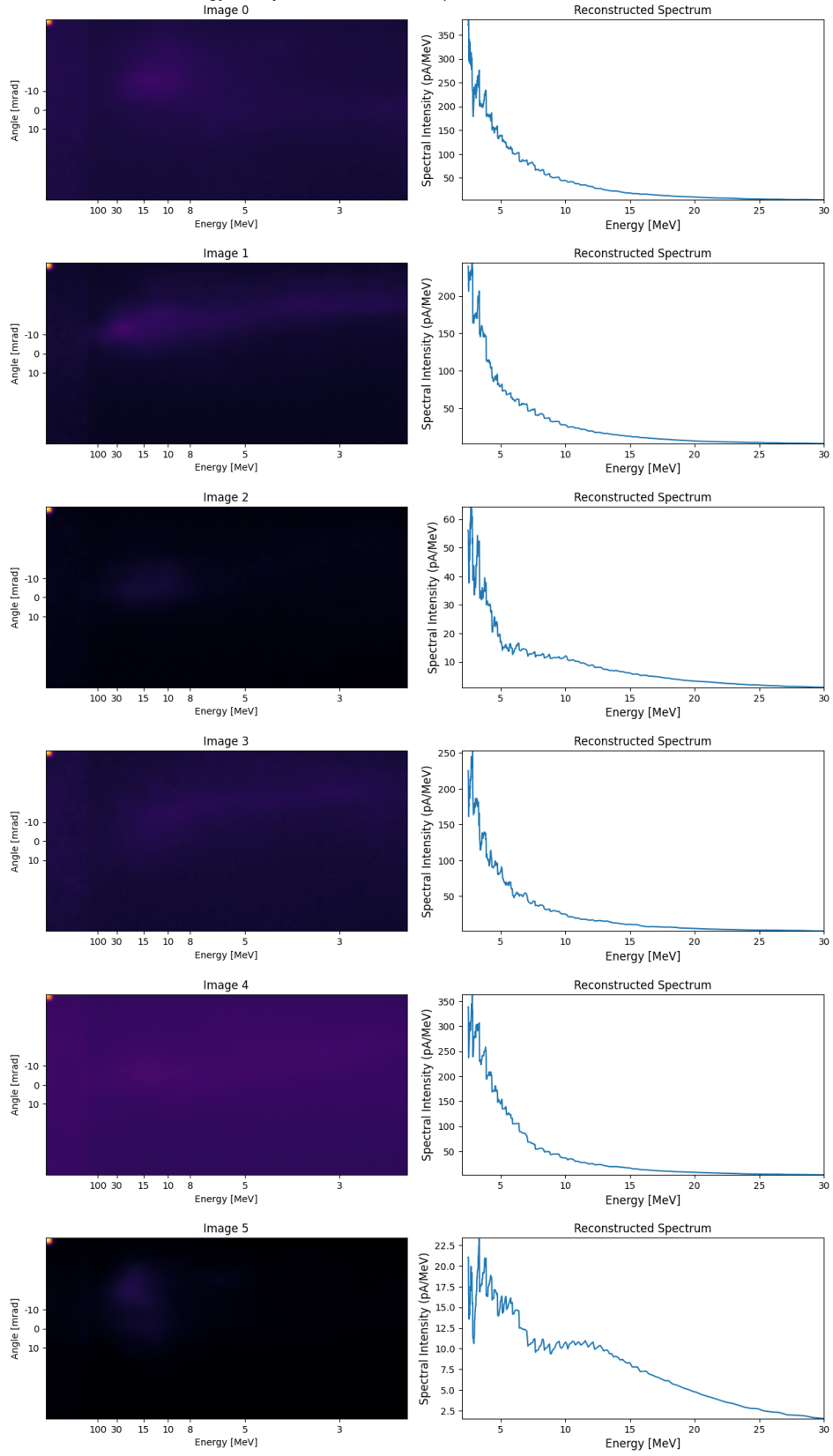
Energy: 32 mJ, Pressure: 15 bar, Acquisition time: 15 ms, Model: 1



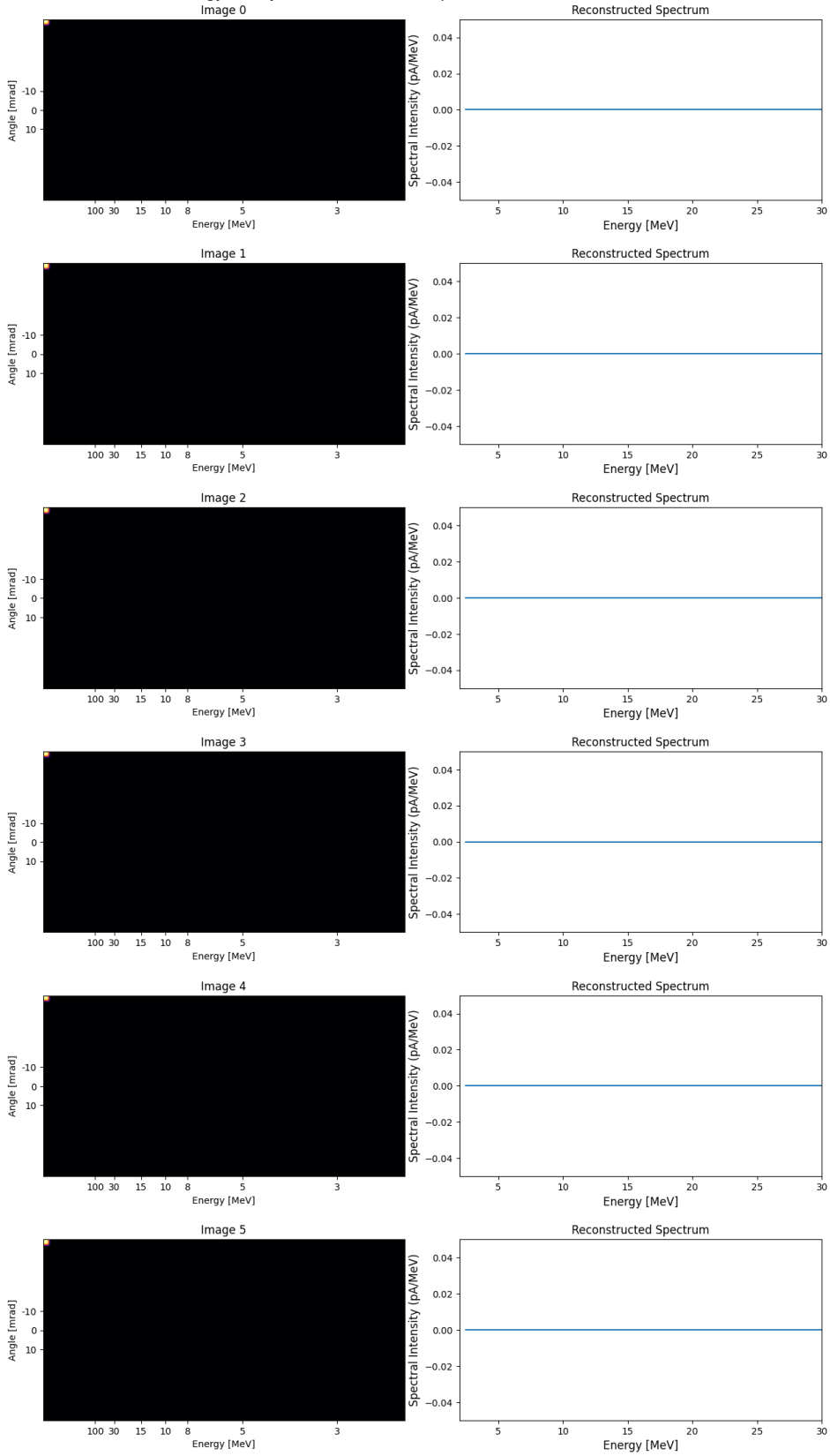
Energy: 32 mj, Pressure: 15 bar, Acquisition time: 15 ms, Model: 2



Energy: 50 mJ, Pressure: 25 bar, Acquisition time: 25 ms, Model: 1



Energy: 50 mJ, Pressure: 25 bar, Acquisition time: 25 ms, Model: 2



Bibliography

1. SOHL-DICKSTEIN, Jascha; WEISS, Eric A.; MAHESWARANATHAN, Niru; GANGULI, Surya. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. 2015. Available from DOI: 10.48550/ARXIV.1503.03585.
2. HO, Jonathan; JAIN, Ajay; ABBEEL, Pieter. Denoising Diffusion Probabilistic Models. 2020. Available from DOI: 10.48550/ARXIV.2006.11239.
3. OPPENLAENDER, Jonas. The Creativity of Text-to-Image Generation. 2022. Available from DOI: 10.1145/3569219.3569352.
4. RAMESH, Aditya; PAVLOV, Mikhail; GOH, Gabriel; GRAY, Scott; VOSS, Chelsea; RADFORD, Alec; CHEN, Mark; SUTSKEVER, Ilya. Zero-Shot Text-to-Image Generation. 2021. Available from DOI: 10.48550/ARXIV.2102.12092.
5. ROMBACH, Robin; BLATTMANN, Andreas; LORENZ, Dominik; ESSER, Patrick; OMMER, Björn. High-Resolution Image Synthesis with Latent Diffusion Models. 2021. Available from DOI: 10.48550/ARXIV.2112.10752.
6. KSHETRI, Nir; DWIVEDI, Yogesh K.; DAVENPORT, Thomas H.; PANTELI, Niki. Generative artificial intelligence in marketing: Applications, opportunities, challenges, and research agenda. *International Journal of Information Management*. 2023, p. 102716. ISSN 0268-4012. Available from DOI: 10.1016/j.ijinfomgt.2023.102716.
7. KUMARI, Nupur; ZHANG, Bingliang; ZHANG, Richard; SHECHTMAN, Eli; ZHU, Jun-Yan. Multi-Concept Customization of Text-to-Image Diffusion. 2022. Available from DOI: 10.48550/ARXIV.2212.04488.
8. OTTEN, Sydney; CARON, Sascha; SWART, Wieske de; BEEKVELD, Melissa van; HENDRIKS, Luc; LEEUWEN, Caspar van; PODAREANU, Damian; RUIZ DE AUSTRI, Roberto; VERHEYEN, Rob. Event generation and statistical sampling for physics with deep generative models and a density information buffer. *Nature Communications*. 2021, vol. 12, no. 1. ISSN 2041-1723. Available from DOI: 10.1038/s41467-021-22616-z.
9. LIU, Ziming; LUO, Di; XU, Yilun; JAAKKOLA, Tommi; TEGMARK, Max. GenPhys: From Physical Processes to Generative Models. 2023. Available from DOI: 10.48550/ARXIV.2304.02637.
10. HUANG, Rongjie; LAM, Max W. Y.; WANG, Jun; SU, Dan; YU, Dong; REN, Yi; ZHAO, Zhou. FastDiff: A Fast Conditional Diffusion Model for High-Quality Speech Synthesis. 2022. Available from DOI: 10.48550/ARXIV.2204.09934.
11. GIBNEY, Elizabeth. How the revamped Large Hadron Collider will hunt for new physics. *Nature*. 2022, vol. 605, no. 7911, pp. 604–607. ISSN 1476-4687. Available from DOI: 10.1038/d41586-022-01388-6.

12. TAJIMA, T.; DAWSON, J. M. Laser Electron Accelerator. *Physical Review Letters*. 1979, vol. 43, no. 4, pp. 267–270. ISSN 0031-9007. Available from DOI: 10.1103/physrevlett.43.267.
13. LAZZARINI, C. M.; GRITTANI, G. M.; VALENTA, P.; ZYMAK, I.; ANTIPENKOV, R.; CHAULAGAIN, U.; GONCALVES, L. V. N.; GRENFELL, A.; LAMAC, M.; LORENZ, S.; NEVRKLA, M.; SOBR, V.; SPACEK, A.; SZUBA, W.; BAKULE, P.; KORN, G.; BULANOV, S. V. 50 MeV electron beams accelerated by a terawatt scalable kHz laser. 2023. Available from DOI: 10.48550/ARXIV.2302.11415.
14. SHALLOO, R. J.; DANN, S. J. D.; GRUSE, J.-N.; UNDERWOOD, C. I. D.; ANTOINE, A. F.; ARRAN, C.; BACKHOUSE, M.; BAIRD, C. D.; BALCAZAR, M. D.; BOURGEOIS, N.; CARDARELLI, J. A.; HATFIELD, P.; KANG, J.; KRUSHELNICK, K.; MANGLES, S. P. D.; MURPHY, C. D.; LU, N.; OSTERHOFF, J.; PÖDER, K.; RAJEEV, P. P.; RIDGERS, C. P.; ROZARIO, S.; SELWOOD, M. P.; SHAHANI, A. J.; SYMES, D. R.; THOMAS, A. G. R.; THORNTON, C.; NAJMUDIN, Z.; STREETER, M. J. V. Automation and control of laser wakefield accelerators using Bayesian optimization. *Nature Communications*. 2020, vol. 11, no. 1. ISSN 2041-1723. Available from DOI: 10.1038/s41467-020-20245-6.
15. BIRDSALL, C.K.; LANGDON, A.B. *Plasma Physics via Computer Simulation*. CRC Press, 2018. ISBN 9781482263060. Available from DOI: 10.1201/9781315275048.
16. ELI BEAMLINES FACILITY. *About ELI Beamlines*. 2024. Available also from: <https://www.eli-beams.eu/about/>. Accessed: 2024-01-10.
17. ELI BEAMLINES FACILITY. *Laser 1: Allegra 100 mJ 1 kHz*. 2024. Available also from: <https://www.eli-beams.eu/facility/lasers/laser-1-allegra-100-mj-1-khz/>. Accessed: 2024-01-10.
18. TAJIMA, T.; YAN, X. Q.; EBISUZAKI, T. Wakefield acceleration. *Reviews of Modern Plasma Physics*. 2020, vol. 4, no. 1. ISSN 2367-3192. Available from DOI: 10.1007/s41614-020-0043-z.
19. FAURE, J.; GLINEC, Y.; PUKHOV, A.; KISELEV, S.; GORDIENKO, S.; LEFEBVRE, E.; ROUSSEAU, J.-P.; BURG, F.; MALKA, V. A laser-plasma accelerator producing monoenergetic electron beams. *Nature*. 2004, vol. 431, no. 7008, pp. 541–544. ISSN 1476-4687. Available from DOI: 10.1038/nature02963.
20. BRÜMMER, Theresa; DEBUS, Alexander; PAUSCH, Richard; OSTERHOFF, Jens; GRÜNER, Florian. Design study for a compact laser-driven source for medical x-ray fluorescence imaging. *Physical Review Accelerators and Beams*. 2020, vol. 23, no. 3, p. 031601. ISSN 2469-9888. Available from DOI: 10.1103/physrevaccelbeams.23.031601.
21. COLE, J. M.; WOOD, J. C.; LOPES, N. C.; PODER, K.; ABEL, R. L.; ALATABI, S.; BRYANT, J. S. J.; JIN, A.; KNEIP, S.; MECSEKI, K.; SYMES, D. R.; MANGLES, S. P. D.; NAJMUDIN, Z. Laser-wakefield accelerators as hard x-ray sources for 3D medical imaging of human bone. *Scientific Reports*. 2015, vol. 5, no. 1. ISSN 2045-2322. Available from DOI: 10.1038/srep13244.
22. CHIU, Charles; FOMYTSKYI, Mykhailo; GRIGSBY, Franklin; RAISCHEL, Frank; DOWNER, Michael C.; TAJIMA, Toshiki. Laser electron accelerators for radiation medicine: A feasibility study. *Medical Physics*. 2004, vol. 31, no. 7, pp. 2042–2052. ISSN 2473-4209. Available from DOI: 10.1118/1.1739301.
23. GIULIETTI, Antonio. *Laser-Driven Particle Acceleration Towards Radiobiology and Medicine*. Ed. by GIULIETTI, Antonio. Springer International Publishing, 2016. ISBN 9783319315638. ISSN 2197-5647. Available from DOI: 10.1007/978-3-319-31563-8.
24. NICKS, B. S.; TAJIMA, T.; ROA, D.; NEČAS, A.; MOUROU, G. Laser-wakefield application to oncology. *International Journal of Modern Physics A*. 2019, vol. 34, no. 34, p. 1943016. ISSN 1793-656X. Available from DOI: 10.1142/s0217751x19430164.

25. CYBENKO, G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*. 1989, vol. 2, no. 4, pp. 303–314. ISSN 1435-568X. Available from DOI: 10.1007/bf02551274.
26. GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
27. AMARI, S.-I. Learning Patterns and Pattern Sequences by Self-Organizing Nets of Threshold Elements. *IEEE Transactions on Computers*. 1972, vol. C-21, no. 11, pp. 1197–1206. ISSN 0018-9340. Available from DOI: 10.1109/t-c.1972.223477.
28. HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long Short-Term Memory. *Neural Computation*. 1997, vol. 9, no. 8, pp. 1735–1780. ISSN 1530-888X. Available from DOI: 10.1162/neco.1997.9.8.1735.
29. STREETER, M.J.V.; COLGAN, C.; COBO, C.C.; ARRAN, C.; LOS, E.E.; WATT, R.; BOURGEOIS, N.; CALVIN, L.; CARDERELLI, J.; CAVANAGH, N.; DANN, S.J.D.; FITZGARRALD, R.; GERSTMAYR, E.; JOGLEKAR, A.S.; KETTLE, B.; MCKENNA, P.; MURPHY, C.D.; NAJMUDIN, Z.; PARSONS, P.; QIAN, Q.; RAJEEV, P.P.; RIDGERS, C.P.; SYMES, D.R.; THOMAS, A.G.R.; SARRI, G.; MANGLES, S.P.D. Laser Wakefield Accelerator modelling with Variational Neural Networks. *High Power Laser Science and Engineering*. 2023, pp. 1–9. ISSN 2052-3289. Available from DOI: 10.1017/hpl.2022.47.
30. KINGMA, Diederik P; WELLING, Max. Auto-Encoding Variational Bayes. 2013. Available from DOI: 10.48550/ARXIV.1312.6114.
31. KÖPPEN, Mario. The curse of dimensionality. In: *5th online world conference on soft computing in industrial applications (WSC5)*. 2000, vol. 1, pp. 4–8.
32. RAISSI, Maziar; KARNIADAKIS, George Em. Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics*. 2018, vol. 357, pp. 125–141. ISSN 0021-9991. Available from DOI: 10.1016/j.jcp.2017.11.039.
33. ZHOU, Tian-Yi; HUO, Xiaoming. Learning ability of interpolating deep convolutional neural networks. *Applied and Computational Harmonic Analysis*. 2024, vol. 68, p. 101582.
34. ANTONIOU, Antreas; STORKEY, Amos; EDWARDS, Harrison. Data Augmentation Generative Adversarial Networks. 2017. Available from DOI: 10.48550/ARXIV.1711.04340.
35. SALIMANS, Tim; GOODFELLOW, Ian; ZAREMBA, Wojciech; CHEUNG, Vicki; RADFORD, Alec; CHEN, Xi. Improved Techniques for Training GANs. 2016. Available from DOI: 10.48550/ARXIV.1606.03498.
36. HEUSEL, Martin; RAMSAUER, Hubert; UNTERTHINER, Thomas; NESSLER, Bernhard; HOCHREITER, Sepp. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. *Advances in Neural Information Processing Systems 30 (NIPS 2017)*. 2017. Available from DOI: 10.48550/ARXIV.1706.08500.
37. STOKEL-WALKER, Chris; VAN NOORDEN, Richard. What ChatGPT and generative AI mean for science. *Nature*. 2023, vol. 614, no. 7947, pp. 214–216. ISSN 1476-4687. Available from DOI: 10.1038/d41586-023-00340-6.
38. AI, Stability. *Generative Models Repository* [<https://github.com/Stability-AI/generative-models>]. 2023.
39. KEBAILI, Aghiles; LAPUYADE-LAHORGUE, Jérôme; RUAN, Su. Deep Learning Approaches for Data Augmentation in Medical Imaging: A Review. *Journal of Imaging*. 2023, vol. 9, no. 4, p. 81. ISSN 2313-433X. Available from DOI: 10.3390/jimaging9040081.
40. ALSAFADI, Farah; WU, Xu. Deep generative modeling-based data augmentation with demonstration using the BFBT benchmark void fraction datasets. *Nuclear Engineering and Design*. 2023, vol. 415, p. 112712. ISSN 0029-5493. Available from DOI: 10.1016/j.nucengdes.2023.112712.

41. WEISS, Karl; KHOSHGOFTAAR, Taghi M.; WANG, DingDing. A survey of transfer learning. *Journal of Big Data*. 2016, vol. 3, no. 1. ISSN 2196-1115. Available from DOI: 10.1186/s40537-016-0043-6.
42. RONNEBERGER, Olaf; FISCHER, Philipp; BROX, Thomas. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015. Available from DOI: 10.48550/ARXIV.1505.04597.
43. SONG, Jiaming; MENG, Chenlin; ERMON, Stefano. Denoising Diffusion Implicit Models. 2020. Available from DOI: 10.48550/ARXIV.2010.02502.
44. PREECHAKUL, Konpat; CHATTHEE, Nattanat; WIZADWONGSA, Suttisak; SUWAJANAKORN, Supasorn. Diffusion Autoencoders: Toward a Meaningful and Decodable Representation. 2021. Available from DOI: 10.48550/ARXIV.2111.15640.
45. BETKER, James; GOH, Gabriel; JING, Li; BROOKS, Tim; WANG, Jianfeng; LI, Linjie; OUYANG, Long; ZHUANG, Yuntang; LEE, Joyce; GUO, Yufei; MANASSRA, Wesam; DHARIWAL, Prafulla; CHU, Casey; JIAO, Yunxin; RAMESH, Aditya. Improving Image Generation with Better Captions. *OpenAI*. 2023.
46. HO, Jonathan; SALIMANS, Tim. Classifier-Free Diffusion Guidance. 2022. Available from DOI: 10.48550/ARXIV.2207.12598.
47. RISSANEN, Severi; HEINONEN, Markus; SOLIN, Arno. Generative Modelling With Inverse Heat Dissipation. 2022. Available from DOI: 10.48550/ARXIV.2206.13397.
48. KARNIADAKIS, George Em; KEVREKIDIS, Ioannis G.; LU, Lu; PERDIKARIS, Paris; WANG, Sifan; YANG, Liu. Physics-informed machine learning. *Nature Reviews Physics*. 2021, vol. 3, no. 6, pp. 422–440. ISSN 2522-5820. Available from DOI: 10.1038/s42254-021-00314-5.
49. JALAS, Sören; KIRCHEN, Manuel; MESSNER, Philipp; WINKLER, Paul; HÜBNER, Lars; DIRKWINKEL, Julian; SCHNEPP, Matthias; LEHE, Remi; MAIER, Andreas R. Bayesian Optimization of a Laser-Plasma Accelerator. *Physical Review Letters*. 2021, vol. 126, no. 10, p. 104801. ISSN 1079-7114. Available from DOI: 10.1103/physrevlett.126.104801.
50. SNOEK, Jasper; LAROCHELLE, Hugo; ADAMS, Ryan P. Practical Bayesian Optimization of Machine Learning Algorithms. 2012. Available from DOI: 10.48550/ARXIV.1206.2944.
51. MOCKUS, Jonas; TIESIS, Vytautas; ZILINSKAS, Antanas. The Application of Bayesian Methods for Seeking the Extremum. *Towards Global Optimization*. 1978, vol. 2, no. 117-129, p. 2.
52. SRINIVAS, Niranjan; KRAUSE, Andreas; KAKADE, Sham M.; SEEGER, Matthias. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. *IEEE Transactions on Information Theory*. 2009, vol. 58, no. 5, pp. 3250–3265. ISSN 1557-9654. Available from DOI: 10.1109/tit.2011.2182033.
53. MAIER, A. R.; MESECK, A.; REICHE, S.; SCHROEDER, C. B.; SEGGBROCK, T.; GRÜNER, F. Demonstration Scheme for a Laser-Plasma-Driven Free-Electron Laser. *Physical Review X*. 2012, vol. 2, no. 3, p. 031019. ISSN 2160-3308. Available from DOI: 10.1103/physrevx.2.031019.
54. SUTTON, Richard S. *Reinforcement learning: An introduction*. [Nachdruck]. Ed. by BARTO, Andrew. Cambridge, Massachusetts: The MIT Press, 2014. A Bradford book. ISBN 0262193981.
55. CAPUANO, Francesco; PECELI, Davorin; TIBONI, Gabriele; CAMORIANO, Raffaello; RUS, Bedřich. TempoRL: laser pulse temporal shape optimization with Deep Reinforcement Learning. 2023. Available from DOI: 10.48550/ARXIV.2304.12187.

56. PASZKE, Adam; GROSS, Sam; MASSA, Francisco; LERER, Adam; BRADBURY, James; CHANAN, Gregory; KILLEEN, Trevor; LIN, Zeming; GIMELSHEIN, Natalia; ANTIGA, Luca; DESMAISON, Alban; KÖPF, Andreas; YANG, Edward; DEVITO, Zach; RAISON, Martin; TEJANI, Alykhan; CHILAMKURTHY, Sasank; STEINER, Benoit; FANG, Lu; BAI, Junjie; CHINTALA, Soumith. PyTorch: An Imperative Style, High-Performance Deep Learning Library. 2019. Available from DOI: 10.48550/ARXIV.1912.01703.
57. RAMPAS, Dominic. *dome272/Diffusion-Models-pytorch*. 2023. Available also from: <https://github.com/dome272/Diffusion-Models-pytorch>.
58. HOUGH, Paul V C. *Method and means for recognizing complex patterns*. Inventor: Paul V C HOUGH. Publ.: 1962. US3069654A. Available also from: <https://patents.google.com/patent/US3069654A/en>.
59. BRADSKI, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*. 2000.
60. BASLER AG. *Gain - Basler Product Documentation* [<https://docs.baslerweb.com/gain>]. 2023. Accessed: 2024.
61. VASWANI, Ashish; SHAZEER, Noam; PARMAR, Niki; USZKOREIT, Jakob; JONES, Llion; GOMEZ, Aidan N.; KAISER, Lukasz; POLOSUKHIN, Illia. Attention Is All You Need. 2017. Available from DOI: 10.48550/ARXIV.1706.03762.
62. LOSHCHILOV, Ilya; HUTTER, Frank. Decoupled Weight Decay Regularization. 2017. Available from DOI: 10.48550/ARXIV.1711.05101.
63. LOSHCHILOV, Ilya; HUTTER, Frank. SGDR: Stochastic Gradient Descent with Warm Restarts. 2016. Available from DOI: 10.48550/ARXIV.1608.03983.
64. KARRAS, Tero; AITTALA, Miika; AILA, Timo; LAINE, Samuli. Elucidating the Design Space of Diffusion-Based Generative Models. 2022. Available from DOI: 10.48550/ARXIV.2206.00364.
65. PERNIAS, Pablo; RAMPAS, Dominic; RICHTER, Mats L.; PAL, Christopher J.; AUBREVILLE, Marc. Wuerstchen: An Efficient Architecture for Large-Scale Text-to-Image Diffusion Models. 2023. Available from DOI: 10.48550/ARXIV.2306.00637.

Contents of enclosed medium

src	Library functions
├─ dataset.py	Dataset preprocessing
├─ diffusion.py	Definition of the noising and sampling process
├─ metrics.py	Helper functions for assessing the quality of models
├─ modules.py	Main implementation of the neural network
├─ utils.py	Various helper functions for plotting and spectra calculations
├─ train.py	The main training script
├─ metrics.py	Evaluation script
├─ metrics.ipynb	Plot and table generation
├─ transfer.ipynb	Script for transferring weights
├─ fid.sh	Script for fid calculation
├─ sample.ipynb	Main image generation script