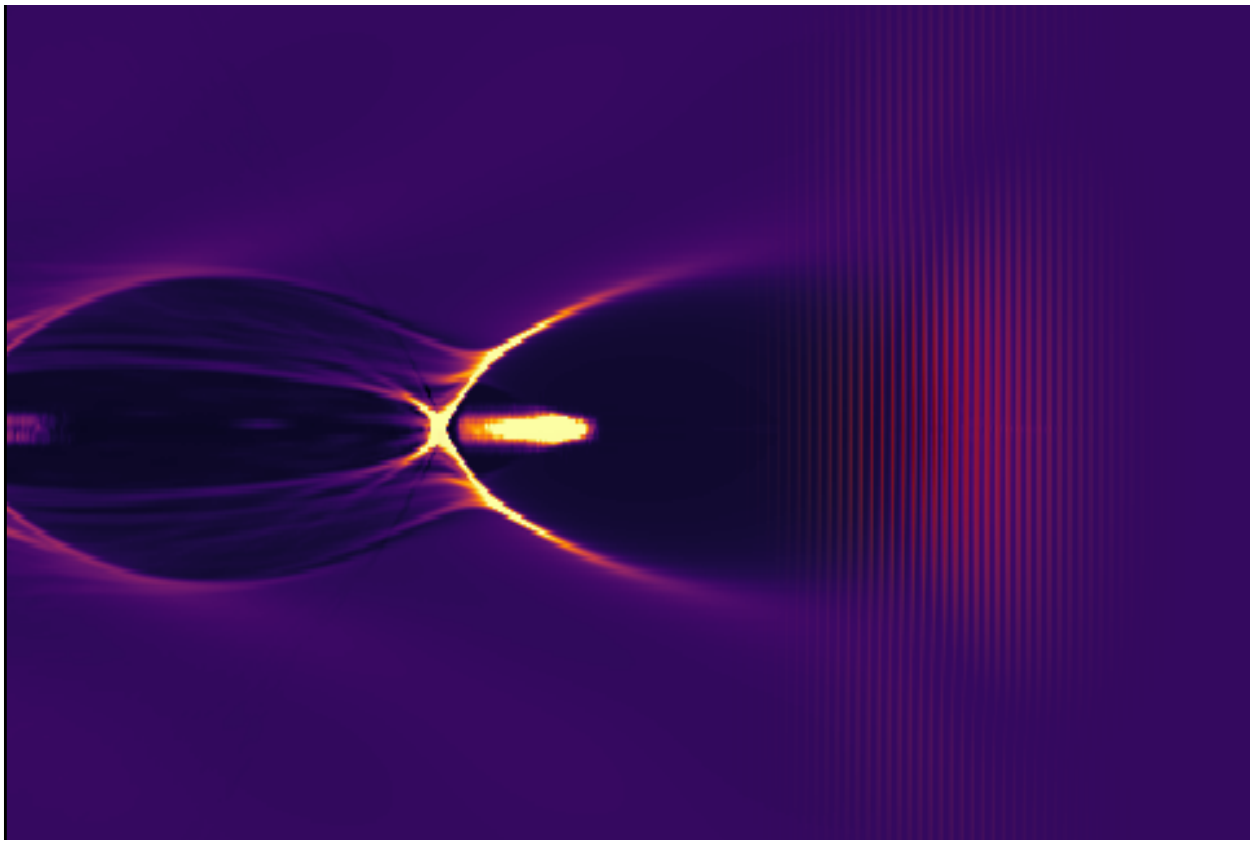

Bayesian optimization of beam quality in plasma accelerated electron beams

A computational study of the performance of Bayesian optimization in noisy environments

Master's thesis in Physics

FRIDA BROGREN
HANNA HALLBORN

DEPARTMENT OF PHYSICS

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021
www.chalmers.se

Master's thesis 2021

Bayesian optimization of beam quality of plasma accelerated electron beams

A computational study of the performance of Bayesian optimization in noisy environments

FRIDA BROGREN
HANNA HALLBORN

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Physics
Division of Subatomic, High Energy and Plasma Physics
Plasma Theory
Chalmers University of Technology
Gothenburg, Sweden 2021

Bayesian optimization of beam quality in plasma accelerated electron beams
A computational study of the performance of Bayesian optimization in noisy environments
FRIDA BROGREN
HANNA HALLBORN

Supervisor: Sören Jalas, Andreas Maier, Deutsches Elektronen-Synchrotron DESY
Examiner: Tünde Fülöp, Department of Physics

Master's Thesis 2021
Department of Physics
Division of Subatomic, High Energy and Plasma Physics
Plasma Theory
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Particle in cell simulation of laser plasma interaction.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2021

Abstract

Laser-plasma acceleration is a promising novel technique for acceleration of charged particles. A challenge with this technique is to maintain a high quality of the accelerated particle bunch. In particular, for accelerated electrons used in Free Electron Lasers, charge and shape of the energy spectrum are important parameters. The aim of this project has been to evaluate and examine the use of Bayesian optimization with respect to these parameters on the LUX laser-plasma accelerator. The focus was to consider how the Bayesian optimization performed under noisy conditions. An important part of Bayesian optimization algorithms is the acquisition function which determines the next point to evaluate in the optimization iteration. In this thesis, two acquisition functions were compared and evaluated from the performance point of view.

In order to test and develop the algorithms, Particle-In-Cell (PIC) simulations were used to emulate the LUX experiment. Further, for cheaper evaluation, a model of the target surface was built from a vast amount of PIC simulated data using Gaussian process regression. With this model, different sampling strategies for each parameter set-point could be investigated. Noise was added to the input parameters as well, yielding a more realistic imitation of the system. A significant improvement was seen when the mean value of 20 input parameters and the mean value of corresponding outputs were fed to the Bayesian optimization algorithm.

Keywords: Gaussian processes, Bayesian optimization, Laser-plasma acceleration, Wakefield acceleration, Noisy Expected Improvement, Expected Improvement.

Acknowledgements

We would like to thank our mentor, Sören Jalas for the support and valuable feedback given during this project. Our supervisor Andreas Maier for doing his best to integrate us in the team at LUX during this trying times of pandemic related limitations. Our examiner Tünde Fülöp for guidance and fruitful discussions. To friends and family for eternal support and much needed breaks from physics studies. Last but not least a big thanks to the whole LUX-team for a warm welcome, great discussions and fun corona-friendly Friday beers over Zoom!

Frida Brogren, Gothenburg, May 2021
Hanna Hallborn, Gothenburg, May 2021

Contents

1
Background

High quality, energetic electron beams are important to various applications such as free electron lasers [1][2][3] and Thomson scattering sources [4] that could be used for medical applications in life science [5]. In particular, free electron lasers (FELs) use relativistic electron beams as gain medium. A small energy spread, that is high spectral brightness amongst the electrons, is necessary for operation [6].

Laser-Plasma Acceleration (LPA) is a promising and currently active research area for acceleration of electrons. It offers a more compact and less expensive way to generate high energy electron beams compared to radio-frequency based acceleration due to several orders of magnitude larger accelerating gradients [7]. LPA uses a plasma target in which a wave is excited by a driver laser. This wave can then trap electrons and accelerate them to several GeV energies in only a few centimeters [8].

DESY (Deutsches Elektronen-Synchrotron) is one of the leading accelerator centers in the world. At the center, research within the fields of particle accelerators, photon science, particle physics and astroparticle physics are conducted. A research project with the aim to demonstrate a plasma-driven FEL is currently conducted at the LUX beamline at DESY. The aim of this master project is to contribute to the stability and quality of the electron beam used to drive the FEL.

The LUX beamline is constructed with the aim of developing a laser plasma accelerator with a steady output that can run for a longer period of time maintaining a high beam quality. The system is designed to enable online feedback for fine-tuning of the electron beam via input parameters of the driver laser and plasma target. The quality of the beam is quantified by the amount of charge and the phase space distribution of the accelerated electrons. The properties of the electron beam are governed by a complex interaction between laser and plasma where the energy spectrum of the electron beam is an output resulting from several input parameters that interact non-linearly and are potentially perturbed by noise in the setup. Thus, for optimization, a non-trivial multidimensional function has to be optimized. For usage of electron beams in FELs a narrow energy spread and high median energy is crucial. Until a year ago the optimization was done more or less manually, but recently the group has started to experiment with different optimization algorithms in order to find optimal input parameters [5].

One major obstacle for many optimization algorithms is noise, which is an unavoidable feature of any experimental setup and specifically laser plasma accelerators. The noise in LPAs is inherent due to the stability of today's high-power lasers. The thesis aims to tackle this problem and enhance the performance of the optimization algorithm currently employed. This is done by reviewing the use and implementation of Baysian optimization (BO) on a collection of parameters that govern the electron acceleration in the LUX beamline. An ideal algorithm should be robust against fluctuations and noise in the experimental setup. The BO algorithms are evaluated through computational studies on multidimensional functions created through PIC simulations and Gaussian process regression modeling. For efficient evaluation, Gaussian process regression modeling is used since PIC simulations are too time demanding. The staring point is the existing BO algorithm developed by Sören Jalas at LUX and earlier research of modeling of the system (see [9]). The aim is to improve this algorithm further to cope with noise in the experimental data and to do this efficiently further research of how to model the target surface in an efficient way is needed.

2
Theory

In this section the physical theory needed to understand plasma acceleration is explained, starting with a brief discussion of general plasma interaction in Sec. 2.1. Thereafter, the discussion will dive more deeply into the physical process of plasma acceleration - the prerequisites and driving forces that make it possible and the specific processes of the system used in this project. Further, in Sec. 2.2 the statistical theory used for modeling and optimization are presented. This entails a thorough description of the theory behind Gaussian processes and the implementation of Bayesian optimization.

2.1 Plasma dynamics

A plasma is a quasi-neutral gas, consisting of charged and neutral particles that exhibit collective behaviour. The motion of charged particles in electromagnetic fields is governed by the Lorentz force. The equations of motion for each particle i in the plasma can be stated as

$$\begin{aligned} \mathbf{v}_i &= \dot{\mathbf{x}}_i, \\ m_i \dot{\mathbf{v}}_i &= q_i(\mathbf{E} + \mathbf{v}_i \times \mathbf{B}), \end{aligned} \tag{2.1}$$

where $\mathbf{E}$ is the electric field strength and $\mathbf{B}$ is the magnetic flux density described by Maxwell's equations

$$\begin{aligned} \nabla \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t}, \\ \nabla \times \mathbf{B} &= \mu_0 \epsilon_0 \frac{\partial \mathbf{E}}{\partial t} + \mu_0 \mathbf{j}, \\ \nabla \cdot \mathbf{E} &= \frac{\rho}{\epsilon_0}, \\ \nabla \cdot \mathbf{B} &= 0. \end{aligned} \tag{2.2}$$

The charge density ρ and current density $\mathbf{j}$ is in turn determined by the location and velocities of all charged particles in the plasma as

$$\rho(\mathbf{x}, t) = \sum_{i=1}^{N} q_i \delta(\mathbf{x} - \mathbf{x}_i(t)) \quad \text{and} \quad \mathbf{j}(\mathbf{x}, t) = \sum_{i=1}^{N} q_i \mathbf{v}_i \delta(\mathbf{x} - \mathbf{x}_i(t)), \tag{2.3}$$

implying a feedback loop for the dynamics of each particle. It is clear that for a plasma, consisting of a large number of charged particles, the full description of the

system above is intractable. Numerical analysis can be made with particle in cell (PIC) simulations where forces on the charged particles due to the EM fields and EM fields due to charge and current densities are calculated at alternating timesteps.

A reduced description of plasma dynamics, where electrons and ions are treated as fluids, can be derived from statistical moments of the single particle distribution $f(\mathbf{x}, \mathbf{v}, t)$ and its temporal evolution in phase space [10]. For plasma acceleration, utilizing high energy lasers, the time scale of the electron motion is much faster than for the heavier ions which can therefore be considered stationary and only the electron fluid equations are required to describe interaction. Furthermore, if thermal velocities of the individual electrons are assumed to be negligible compared to motions induced by the electromagnetic fields, then the system of fluid equations is closed since the pressure can be neglected [10]. This yields the *relativistic cold fluid equations*, consisting of the continuity equation

$$\frac{\partial n_e}{\partial t} + \nabla \cdot (n_e \mathbf{v}) = 0 \tag{2.4}$$

and fluid momentum equation

$$\frac{\partial \mathbf{p}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{p} = -q \left[\mathbf{E} + \frac{(\mathbf{v} \times \mathbf{B})}{c} \right]. \tag{2.5}$$

Here $n_e = n_e(\mathbf{x}, t)$ is the electron density, $\mathbf{v}(\mathbf{x}, t)$ is the fluid velocity and $\mathbf{p} = \gamma m \mathbf{v}$ is the fluid momentum with relativistic gamma factor

$$\gamma = \sqrt{1 + \left(\frac{p}{mc}\right)^2} = \frac{1}{\sqrt{1 - \beta^2}}. \tag{2.6}$$

The last equality comes from defining $\beta = |\mathbf{v}|/c$ as the normalized fluid velocity. The cold electron fluid equations together with Maxwell's equations describe the macroscopic dynamics of plasma interacting with intense electromagnetic fields. In laser-plasma acceleration, this can be used to qualitatively describe the excitation of the plasma wave which is referred to as the wakefield.

2.1.1 Laser-Plasma Acceleration (LPA)

In this section we will explain the underlying processes resulting in accelerated electrons. If no other reference is made or facts are explicitly proven the content is based on the work of A. R. Maier et al. in *Brilliant Light Sources driven by Laser-Plasma Accelerators* Sec. 1.2 (see [6]).

Even though the full 3D description of a plasma that interacts with a high energy laser is intractable, some of the features of the system can be understood heuristically if some simplifications by reasonable assumptions are made. The process of laser plasma acceleration can roughly be divided into three stages: (1) the generation of the wakefield from electron density modulation induced by the laser, (2) injection of electrons into the wakefield (ionization and trapping process) and (3) acceleration of the trapped electrons, that is energy transfer from the plasma wave to the electron beam.

2.1.1.1 Wakefield generation

To understand plasma acceleration it is useful to imagine the motion of loose buoy at sea. When a wave approaches, the buoy will accelerate forward by the push of the wave, on the top of the wave it will start accelerating backward sliding down the wave and eventually ending up at a point slightly displaced from the initial position. A similar system is the electron exposed to an electromagnetic Gaussian pulse. Here the intensity of the pulse represents the height of the wave. The fast oscillations of the electromagnetic field create a force similar to the push of the wave: the ponderomotive force. This force is essentially what drives the wave motion in a plasma.

Before describing the effects and characteristics of this force it is prudent to outline the concepts of the *linear* and *nonlinear regime* of laser-plasma interaction. These regimes are characterized by increasing electron velocities. In the nonlinear regime the motion of electrons can no longer be treated non-relativistically and the equations describing the interaction become nonlinear. The nonlinearity of the interaction scales with the strength of the driver laser, therefore one usually characterizes the interaction on the basis of a_0 which is the peak amplitude of the normalized vector potential $\mathbf{a} = e\mathbf{A}/m_e c$. If $a_0 \ll 1$ the motion can be approximated to be linear, in contrast if $a_0 > 1$ the motion is nonlinear.

In the linear regime the ponderomotive force can be derived from the fluid momentum equation (2.5) by expressing $\mathbf{B}$ and $\mathbf{E}$ in the magnetic vector potential $\mathbf{A}$ and the electromagnetic potential Φ and assuming that the laser is transversely polarized [11]. After separating the motion and fields into one fast oscillating quiver motion and one slowly varying motion one arrives at the expression $F_p = -m_e c^2 \nabla \mathbf{a}^2/2 \propto -\nabla I$, where I is the intensity of the laser [11]. The ponderomotive force can therefore be seen as a radiation pressure of the laser. In the non-linear limit an analytic treatment of the system is not possible and one often relies on numerical and approximate solutions. For further analytical reasoning we are going to employ the 1D approximation of the laser-plasma interaction, which is valid for sufficiently large spot size of the laser [12]. Within this framework the longitudinal component of the fluid momentum equation can be written as (see App. A.1 for derivation)

$$\frac{d\mathbf{p}_{||}}{dt} = e\frac{\partial \Phi}{\partial z} - \frac{e}{2m_e c\gamma}\frac{\partial \mathbf{A}^2}{\partial z} \tag{2.7}$$

suggesting that the time evolution of the plasma fluid momentum is driven by the electric potential induced by charge separation and the ponderomotive force from the laser proportional to $\partial \mathbf{A}^2/\partial z$, i.e. once more proportional to the gradient of the intensity.

When a laser pulse enters a plasma, the electrons are pushed forward by the described ponderomotive force. This creates a positively charged area behind the wave, remember that the ions are assumed to be immobile on this timescale due to their larger mass. Electrons placed on the peak of the intensity or slightly behind will feel a pull backwards towards this positive area, the force backwards will be increased by the ponderomotive force pushing the electrons away from the higher intensity

of the pulse. These two forces combined will cause the electrons to overshoot their initial position and create a wave-like motion that trails after the laser pulse. The electric field that is created by the charge separation due to this electron density modulation is called a wakefield.

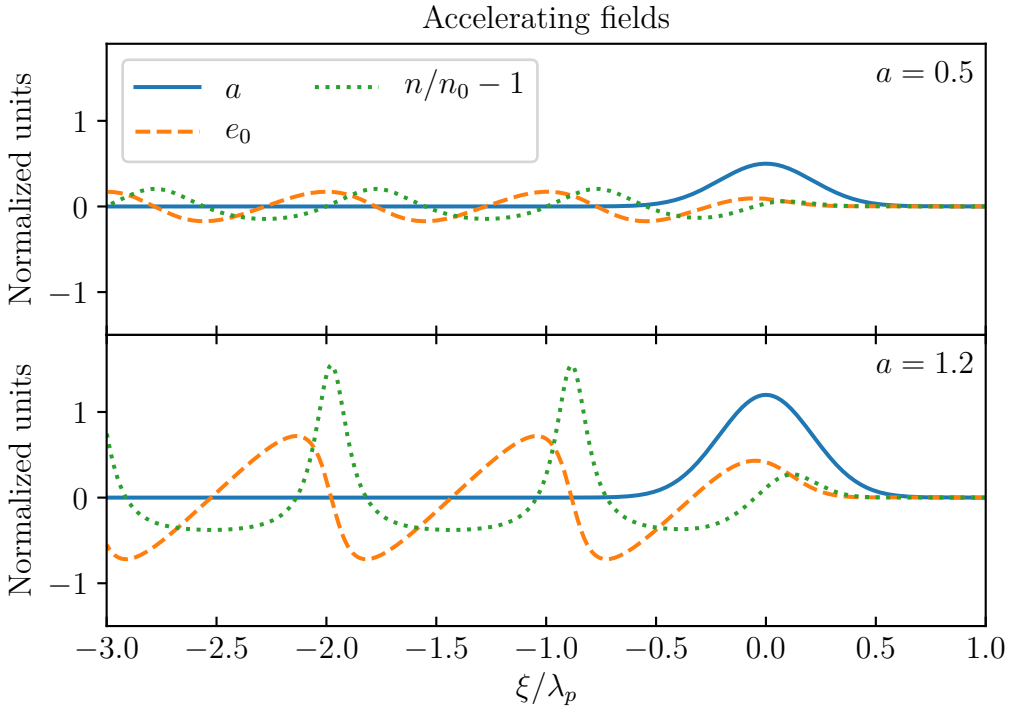

Figure 2.1: Electric fields in the linear ($a < 1$) and non-linear regime ($a > 1$). The electric field e_0, density profile $n_e/n_0 - 1$ and vector potential are all normalized. For derivation of these fields see Sec. 2.1.1.2.

In the linear regime ($a_0 \ll 1$) the term on the right hand side of Eq. (2.5) is $\mathbf{v}\times\mathbf{B} \approx 0$, which leads to a linear motion. In this regime the frequency of the wakefield ω_{WF} is the plasma frequency $\omega_{WF}^2 = \omega_p^2 = n_0 e^2/\epsilon_0 m_e$ which gives a wavelength equal to the plasma wavelength $\lambda_{WF} = \lambda_p = c2\pi/\omega_p$. The wavelength is thus inversely proportional to the square root of the density of the plasma. Accordingly, a lower density will elongate the plasma wave. An increase in the intensity of the driver laser ($a_0 > 1$) will make the velocities of the electrons relativistic and the equation of motion nonlinear, i.e. $\mathbf{v}\times\mathbf{B} \neq 0$ and $\mathbf{p} = \gamma m_e \mathbf{v}$. This in turn causes the relativistic mass of the electrons to increase and thus the accelerating field to elongate and adapt a saw-tooth shaped form. The fields in the linear and non-linear regime are depicted in Fig. 2.1. The maximum acceleration length is limited by the *dephasing-length* which scales with λ_p and is the maximum length an electron is experiencing an accelerating force in the wakefield in the laboratory frame. Therefore, the wavelength of the wake is an important parameter when considering how much energy an electron can gain from a plasma wave.

For strongly driven plasma waves ($a_0 \ll 1$) the electrons can locally move faster than the phase velocity of the wave causing an effect called *wave breaking*, similar to the wave breaking of water. At this point the fluid theory breaks down and one must employ numerical methods to describe the dynamics. This phenomenon does not only happen in the longitudinal direction but also in the transversal. For strong laser intensities the electrons are pushed transversely outwards by the laser pulse creating a depletion area behind the driver pulse. The electrons are then streaming around this bubble-like structure and meet at the back of the bubble. This regime is called the *bubble* or *blow-out regime* and is depicted in Fig. 2.2 generated by PIC simulations.

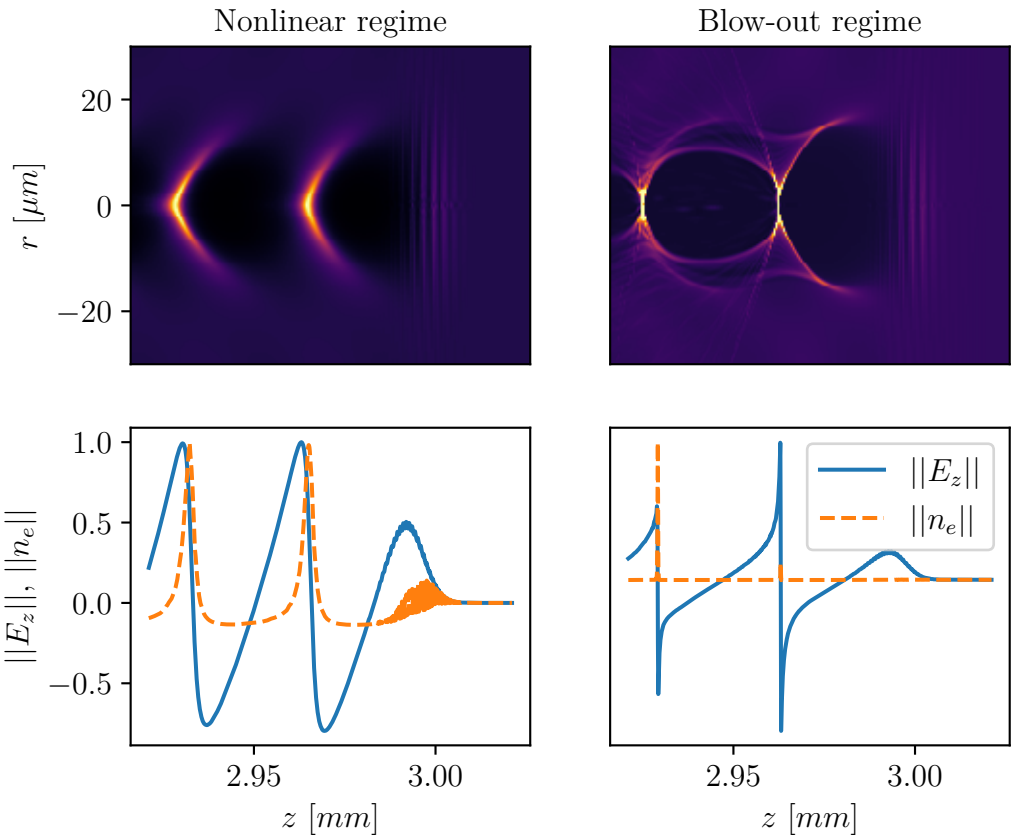

Figure 2.2: PIC simulation of wakefield induced by a laser propagating in a plasma with plasma density $n_0 = 12 \cdot 10^{18}$ cm^{-3}. To the left is a simulation with normalized vector potential $a = 1.7$ resulting in a nonlinear wakefield and to the right is a simulation with $a = 2.6$ where the blow-out regime is visible. The upper figures show 2D slices along the z-axis of the electron density n_e. The lower figures show n_e and the accelerating field E_z along the z-axis at $r = 0$. $||\cdot||$ denotes normalization to the largest value.

2.1.1.2 Electron injection

So far, we have only discussed the collective wave motion of the plasma. The acceleration however is the consequence of a relatively small number of electrons breaking

free from this collective motion and starts following the wake as it moves through the plasma. This process is called electron injection.

For successful trapping the accelerated electrons must be confined in the transversal direction as well as in the longitudinal direction. Qualitatively, for the transversal confinement, one can say that the positively charged area after the laser, the depletion area, attracts electrons keeping them focused towards the center of the wake. If the electron slips backward the effect is the opposite, it enters an area of high electron densities which pushes the electron away from the center.

The longitudinal trapping can be described by studying equipotential lines in phase space. The Hamiltonian of a relativistic particle in an electromagnetic potential is

$$H = \sqrt{(m_e c^2)^2 + (|\mathbf{p} - q\mathbf{A}|^2)c^2} - q\phi \tag{2.8}$$

where ϕ is the electromagnetic potential and $\mathbf{A}$ is, again, the magnetic vector potential. Note that $\mathbf{p}$ is now the single particle momentum of a particle in the electromagnetic potential created by the collective momentum of the plasma. By making a Lorentz transformation $\xi = \gamma(z - tv_p)$, it is possible to describe the dynamics in a reference system moving with the laser pulse. The velocity of the driver laser is v_p which is also equal to the phase velocity of the plasma wave. In an electromagnetic field the potential transforms as $\phi' = \gamma(\phi - v_p A_z)$ and the kinetic energy transforms as $E' = \gamma(E + v_p p_z)$ under a Lorenzt transformation in z. When only considering motion in the longitudinal z direction and choosing coulomb gauge $\nabla \cdot \mathbf{A} = 0, A_z = 0$ this yields [13]

$$H' = \gamma\left(\sqrt{(m_e c^2)^2 + p_z^2 c^2 - q^2|\mathbf{A}|^2 c^2} - q\phi' - v_p p_z\right). \tag{2.9}$$

If we assume that the fields ϕ' and $\mathbf{A}$ are static in this frame, i.e. non-evolving, then the Hamiltonian is conserved and this boosted Hamiltonian can be used to plot equipotential lines. By normalizing H' as $h_0 = H'/\gamma m_e c^2$

$$h_0 = \sqrt{1 + \frac{p_z^2}{m_e^2 c^2} - q^2\frac{|\mathbf{A}|^2}{m_e^2 c^2}} - q\frac{\phi'}{m_e c^2} - v_p\frac{p_z}{m_e c^2} \tag{2.10}$$

introducing new normalized quantities $u_z = p_z/m_e c$, $q|A|/m_e c = a$ and $q\phi'/m_e c^2 = \Phi$ we get

$$h_0 = \sqrt{1 + u_z^2 - a^2} - \Phi - \beta_p u_z \tag{2.11}$$

where $\beta_p = v_p/c$. The expression for u_z becomes

$$u_z = \frac{h_0 + \Phi}{1 - \beta_p^2}\left[\beta_p \pm \sqrt{1 - \frac{(1 - \beta_p^2)(1 + a^2)}{(h_0 + \Phi)^2}}\right]. \tag{2.12}$$

Using the cold fluid electron equations (2.4) and (2.5) the non-linear Poisson's equation in the limit $v_p \to c$ gives an expression for Φ (for derivation see [12][14])

$$\frac{\partial^2 \Phi}{\partial \xi^2} = \frac{k_p^2}{2}\left(\frac{1 + a^2}{(1 + \Phi)^2} - 1\right), \tag{2.13}$$

where $k_p = \omega_p/c$. This differential equation can be solved numerically by assuming that the driver laser is non-evolving and

$$a = a_0^2 e^{-8ln(2)k_p^2\xi^2/\pi^2}. \tag{2.14}$$

The factor in the exponent is a scaling of the width of the pulse in longitudinal space such that full width half max is $\lambda_p/2\sqrt{2}$ and is chosen for effective excitation of the wakefield. By using the expression for the Hamiltonian (2.11) and the momentum equation (2.12) the single particle orbits in phase space can be calculated given appropriate values for a_0, v_p and ω_p. As previously stated, the plasma wave travels with the group velocity of the laser. From the dispersion relation for transversal electromagnetic waves in plasmas ($\omega^2 = \omega_p^2 + c^2k^2$) an expression for v_p can be calculated as $v_p = v_{g,laser} = \partial\omega/\partial k \approx c(1 - \omega_p^2/2\omega^2)$, where $\omega = 2\pi c/\lambda$ is the frequency of the laser. The operational wavelength, λ, for high-power lasers in today's LPAs is typically $\sim$ 800 nm [6]. The velocity of the plasma wave is thus a function of the plasma density. Particle trajectories with $a_0 = 1.2$ and $v_p = 0.97c$, corresponding to a plasma density of $n_0 = 9.9 \cdot 10^{19}\,\mathrm{cm}^{-3}$, are shown in Fig. 2.3. Note that the ξ-coordinate is the space coordinate in the boosted frame but the momentum coordinate u_z is proportional to the momentum in the lab-frame.

In the lower left plot of Fig. 2.3 are the electron trajectories in (normalized) phase space. The blue lines are trajectories of electrons that are faster than the wake, these will outrun the wake and are usually unpopulated. The red trajectories are electrons that are too slow, these will be passed by the wake. The green trajectories are trapped electrons that will follow with the wake. For an electron to become trapped it is, as one can see, necessary that the particle has the right phase, this process is called electron injection. There are several techniques of electron injection. In the LPA system described in this thesis *ionization injection* is used. The benefit of this type of injection is that it is in general more stable than other conventional methods. The theory behind ionization injection is to "create" electrons at zero momentum at the bottom of the wake potential. In the right plot in Fig. 2.3 the green lines marks trapped particle orbits and one can see that by injecting electrons at $u_z = 0$ it is possible to effectively trap electrons in the wake. In practice these electrons are created by further ionization of the plasma medium at the peak intensity of the driving laser. As one can see in the right graph of Fig. 2.3 at the peak of the laser ($\xi/\lambda_p = 0$) there can be no ionization injection in this case since the trapped particle orbits do not cross the zero-momentum at that point. However, by increasing the velocity of the plasma wave, i.e. decreasing the plasma density, and increasing the driver laser the zero-momentum boundary is moved forward. Typical LPAs work with plasma densities of 10^{15} to $10^{19}\mathrm{cm}^{-1}$ [6] which corresponds to a plasma wave group velocity of $v_p \in [0.9971296c, 0.9999997c]$ and with intensities corresponding to $a_0 =$ 2-4. One should be aware that the analysis presented in this section is an approximation of an idealized one-dimensional system in the cold fluid limit. In three-dimensional space the dynamics become much harder to approximate analytically. One therefore often relies on numerical simulations like PIC simulations (see Sec. 4.1) for description of the dynamics, e.g the simulation seen in Fig. 2.2.

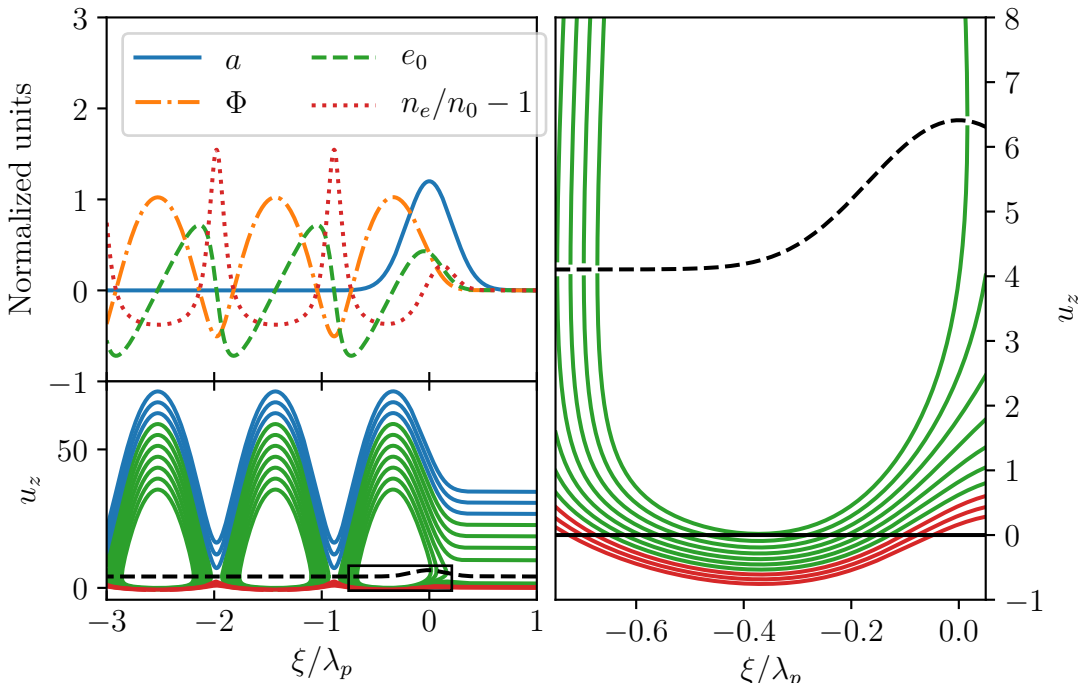

Figure 2.3: Fields and particle orbits in a boosted frame that moves at the same speed as the plasma wake. The plot in the upper left corner shows the normalized fields, the plot in the lower left corner shows the particle orbits in phase space. The black dashed lines are where the two solutions of (2.12) meet. The right plot shows a zoomed-in section of the lower left plot, this section is marked with a black box. The figure was made by setting $a_0 = 1.2$, $v_p = 0.97c$. The normalized Hamiltonian h_0 ranges between 0 to 1. The density was calculated by $n_e/n_0 - 1 = \frac{1}{k_p^2}\frac{\partial^2\Phi}{\partial\xi^2}$ and the normalized E-field $e_0 = -\frac{1}{k_p}\frac{\partial\Phi}{\partial\xi}$.

2.1.1.3 Acceleration at the LUX experiment

In the calculations of the accelerating fields (in Sec. 2.1.1.2) the existence of the injected electron bunch has not been taken into account. The force from the space charge of a relativistic electron bunch can displace plasma electrons in a similar way as the driver laser. The electron bunch has a much larger relativistic mass than the plasma electrons which are thus pushed away by the electron bunch inducing another wakefield, out of phase with the wake driven by the laser [7]. The wakefield induced by the electron bunch will thus affect the accelerating field generated by the laser pulse, this effect is called *beam loading*. It is possible to completely flatten the accelerating field locally by injecting electrons with a specific density profile (or current profile in the lab frame) [13]. A flat accelerating field is crucial to maintain a small energy spread of the accelerated electron bunch [13]. This can be controlled partly by localization of the ionized gas. The process of beam loading is strongly affected by the composition of the gas forming the plasma and the phase of the injected electrons.

In the LUX setup (see Fig. 2.4) the laser pulse both ionizes the gas as well as induces the accelerating plasma wave. The gas target consists of a capillary tube with three gas inlets. In the first tube a mixture of nitrogen, N_2, and hydrogen, H_2, gas is injected, in the second and third pure hydrogen gas. Before the laser enters the target the electron density is zero since all electrons are still bound to the gas molecules. The front of the laser, where the energy is relatively small, fully ionizes the hydrogen gas, i.e. two electrons per hydrogen molecule are added to the electron density and a plasma is formed. For the nitrogen gas, only the outer shells with five electrons are ionized at the lower intensities of the front end of the laser. At the peak intensity of the laser however, inner K-shell electrons of the nitrogen are also ionized (through tunneling ionization) providing the extra electrons at ~ 0 momentum. If the peak intensity is at a proper phase of the wake, as discussed in sec.2.1.1.2, these electrons are trapped and can be accelerated.

To achieve optimal beam loading both the driver laser properties and the plasma density profile must be tailored. A crucial part is to have injection of electrons along the plasma density downramp seen in Fig. 2.4. The decrease in plasma density elongates the wake which in turn lowers the threshold for ionization injection [9]. The combination of lower density and less nitrogen doped gas gives an injected beam with a triangular density profile that is increasing upstream [9].

There will be a small positive correlation in the initial phase space of the electron bunch, right after the injection, since electrons will not be injected exactly at the same time [9]. This gives the electrons that are injected first a lead in experienced acceleration and generates an energy spread. At the LUX experiment the nitrogen gas is constrained to a short region at the beginning of the target such that the electron injection occurs only for a short period of time along a short distance to minimize the positive correlation. This results in a small initial spread in phase space of the electron beam. The small lead is further reduced when the wakefield enters the plasma plateau where the laser reaches higher intensity due to self focusing and

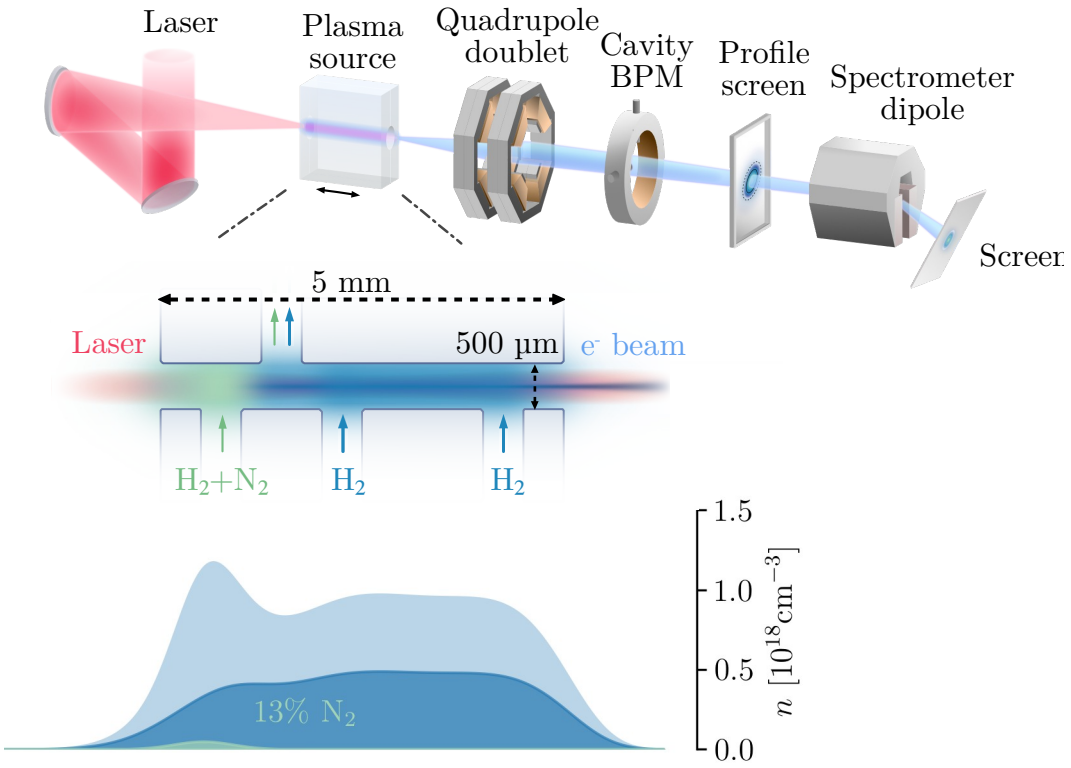

Figure 2.4: Graphical scheme of the LPA. The electrons are accelerated at the plasma source where a mixture of nitrogen and hydrogen is injected through three inlets. Downstream from the plasma source are apparatus for focusing the electron beam in space (the quadrupole doublet) and diagnostics for measuring energy spectra and density. The cavity beam position monitor (BPM) measures the transversal position of the beam and the charge of the electron beam. The profile screen measures the transversal distribution and can be removed from the beamline to allow measurement of the energy spectra, which is done by the spectrometer dipole and the last screen. Figure from [9].

a stronger blow-out regime is reached [9]. This yields a strong accelerating field at the back of the wake [9] and the electrons there will catch up with the electrons in the front reducing the energy spread. The small energy spread is then maintained throughout the acceleration if proper beam loading is achieved and the accelerating field is flat.

From the discussion so far it can be concluded that the accelerating process is affected by the density profile of the plasma, the nitrogen concentration, laser focus position and laser energy. Balancing all these parameters is necessary to get an electron bunch with high quality. For instance, the concentration of nitrogen at the beginning of the target will affect both the ionization rate and the trapping process [9]. The ionization rate is increased by higher nitrogen concentration since this results in more nitrogen molecules available for ionization. But the nitrogen concentration also alters the plasma density, n_0, at the beginning of the target as seen in Fig. 2.4 which strongly affects the trapping process [9]. Higher concentra-

tion yields a higher peak in the plasma density and the elongation of the wake at the downramp becomes more prominent. The plasma density at the plateau where the acceleration occurs is determined by the hydrogen density and it can be shown that the accelerating field scales as $\sqrt{n_0}$ [7]. A higher hydrogen density will thus in general yield a higher final energy of the accelerated electron bunch [9]. Too large plasma density will however shorten λ_p and thus the dephasing-length, decreasing the acceleration length for the electrons as discussed before. Further, the amount of charge, i.e. the amount of electrons, that can be accelerated in a single wakefield is limited by the beam loading effect [7]. Too much charge will overload the beam meaning that the wakefield generated from the electron bunch will be so strong that it deforms the laser induced wakefield completely.

Clearly, the output characteristics of the electron bunch are a complex non-linear function of the parameters that govern the processes that create it. The objective of this project and the beamline at LUX is to generate stable electron bunches with small energy spread, high median energy and a large number of electrons i.e. high total charge. This is done by carefully modifying the prerequisites for wakefield excitation, electron injection and beam loading via the properties of the driver laser, specifically the longitudinal focus position and the energy, and the composition of gases in the target.

2.2 Modeling and Optimization

The discussions so far have demonstrated the complexity of a system consisting of a high power laser interacting with a plasma target. It is clear that predicting the properties of electrons accelerated via this process is a very hard task and thus finding optimal input parameters for the desired electron properties is indeed a huge challenge. To tackle this problem Bayesian optimization (BO) can be used which is a global optimization algorithm suitable for optimization of expensive to evaluate black-box functions [15]. The system at LUX is expensive in the way that it takes approximately 1 minute to change working point due to the response of the gas flow settings. Further the pulsed laser has a 1 Hz repetition rate limiting the amount of data that can be obtained at one working point within a reasonable time frame.

For the optimization, the measured outputs from the experiment are summarized and combined to form a target function $f(\mathbf{x})$ that expresses the beam quality which we want to optimize. For the experiment at LUX the target function is composed of the median energy of the accelerated electrons $\tilde{E}$, median absolute deviation ΔE i.e. a measure of the energy spread, and the total charge of the beam Q as

$$f(\mathbf{x}) = \frac{\sqrt{Q}\tilde{E}}{\Delta E}, \tag{2.15}$$

The input variables $\mathbf{x}$ are points in the four-dimensional input parameter space defined by measures of laser focus position, laser energy, concentration ratio of N_2, and plasma density. More complex functions could also be used, this one is chosen since

it is composed of quantities that are easily measured in the experiment. The functional form of the true target function in Eq. (2.15) is unknown, i.e. it is a black-box function, only a few noisy values are known at the measured points. Some knowledge about the surface can be obtained by simulating it through PIC simulations, however the surface may change in unexpected ways depending on changes in the system and it is therefore hard to extrapolate the knowledge from PIC simulations to the experimental setup.

The Bayesian optimization algorithm is divided into two parts. One where the target function is modeled as a probability distribution of possible target functions, conditioned on data points. The second part of Bayesian optimization is to formulate a so called acquisition function to guide the sampling in a more efficient way than the random approach. The modeling of the target function is done through Bayesian inference described in Sec. 2.2.1. In this inference, a Gaussian Process (GP) is used as prior distribution of target functions, Sec. 2.2.2 presents the definition and properties of Gaussian processes and in Sec. 2.2.2.1 the use and properties of kernel functions are discussed. The likelihood of the Bayesian inference and modeling of measurement errors are discussed in Sec. 2.2.3. Model selection in terms of hyperparameter optimization is briefly described in Sec. 2.2.4 and the modeling part ends with Sec. 2.2.5 about making predictions where we arrive at the posterior distribution that represents our model for the target function. The general BO algorithm, utilizing the model and an acquisition function, is presented in Sec. 2.2.6. Theory concerning the specific class of *improvement-based* acquisition functions, which are the focus of this thesis, are then discussed in Sec. 2.2.7.

Table 2.1: Explanation of the variables used in the Sec. 2.2.1.

Variable	Explanation
$\mathbf{x}$	Input data point
$\mathbf{x}_p$	Input prediction point
y	Output data point
$f = f(\mathbf{x})$	Model function value at data point (random variable)
$f_p = f(\mathbf{x}_p)$	Model function value at predictive point (random variable)
$m(\mathbf{x})$	Mean function of GP
$k(\mathbf{x}, \mathbf{x}')$	Covariance function of GP
$\mathbf{K}$	Covariance matrix $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$
$\mu(\mathbf{x})$	Mean function of the posterior predictive
ε	Noise term (random variable)
$\mathbf{e} = [\varepsilon(\mathbf{x}_1), ..., \varepsilon(\mathbf{x_n}))]$	Vector of ε
θ	Vector of hyperparameters e.g $\theta = [l_1, l_2, .., l_n, \sigma_\epsilon, ..]$

2.2.1 Model definition and Bayes theorem

To relate input variables $\mathbf{x}$ to observations y it is possible to specify a model $\mathcal{M}$ which is a procedure for generating observations, including measurement noise ε. In

this case the model consists of the target function $f(\mathbf{x})$ and a noise term

$$\mathcal{M} : y(\mathbf{x}) = f(\mathbf{x}) + \varepsilon. \tag{2.16}$$

The model is in our case fully specified by the function $f(\mathbf{x})$ and the noise ε and the aim is to make inference about these quantities. In Bayesian inference, the prior knowledge about $f(\mathbf{x})$ and ε is also included in the model concept and incorporated through *prior probability distributions*. From Bayes theorem it is possible to express a probability for the considered model given the collection of observations $\mathbf{y} = \{y_i\}_{i=1}^n$ and corresponding inputs $\{\mathbf{x}_i\}_{i=1}^n$ as

$$P(f, \varepsilon|\mathbf{y}, \mathcal{M}) = \frac{P(\mathbf{y}|f, \varepsilon, \mathcal{M})P(f, \varepsilon|\mathcal{M})}{P(\mathbf{y})}, \tag{2.17}$$

where $P(\mathbf{y}|f, \varepsilon, \mathcal{M})$ is the likelihood, $P(f, \varepsilon|\mathcal{M})$ is the prior and $P(\mathbf{y})$ is the evidence (or marginal likelihood). The notation $\mathcal{M}$ will further be dropped since this will be the always present and underlying model and assuming f and ε to be independent yields

$$P(f, \varepsilon|\mathbf{y}) = \frac{P(\mathbf{y}|f, \varepsilon)P(f)P(\varepsilon)}{P(\mathbf{y})}, \tag{2.18}$$

where $P(f, \varepsilon|\mathbf{y})$ is the joint posterior probability distribution for $f(\mathbf{x})$ and ε given data $\mathbf{y}$.

For (linear) parametric regression one would parametrize $f(\mathbf{x}) = \mathbf{w}\Phi(\mathbf{x})$ with $\Phi(\mathbf{x}) = [\phi_1(\mathbf{x}), ..., \phi_M(\mathbf{x})]$, where $\{\phi_m(\mathbf{x})\}_{m=1}^M$ is a set of known basis functions, and perform inference over the weights $\mathbf{w}$. However, in this study non-parametric regression is used keeping $f(\mathbf{x})$ as a somewhat abstract entity namely a probability distribution of functions. As a prior for $f(\mathbf{x})$ a Gaussian process will be used, that is $P(f) = \mathcal{GP}$, which will be introduced in the next Sec. 2.2.2.

2.2.2 Gaussian processes

To describe the prior probability distribution of functions $f(\mathbf{x})$ we use a *Gaussian process* (GP) defined as follows [16]

Definition 2.2.1 *A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution*

The functions are distributed as a Gaussian process according to

$$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right) \tag{2.19}$$

which can be represented as a vector of function values $\mathbf{f} = [f_1, .., f_n]$ at input variables $\{\mathbf{x}_i\}_{i=1}^n$. The vector $\mathbf{f}$ is a collection of random variables, as in Def. 2.2.1, and thus jointly distributed as a multivariate normal distribution according to

$$\mathbf{f} \sim \mathcal{N}(\mathbf{m}, \mathbf{K}). \tag{2.20}$$

Here $\mathbf{m}$ is a vector of mean function values $m_i = m(\mathbf{x}_i)$, and $\mathbf{K}$ is a covariance matrix with elements determined by the covariance function $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.
The reason that we can link the finite collection of function values $\mathbf{f}$ (a collection of random variables) to the continuous functions $f(\mathbf{x})$ is because a collection of random variables implies a consistency condition (marginalization property) which in extension leads to the property that the joint distribution of a finite collection of variables is not affected if more (possibly infinitely many) variables are included in the distribution.

A Gaussian process is a quite abstract concept. One way to understand it is to picture it as an infinite-dimensional Gaussian distribution where every axis corresponds to an input coordinate $\mathbf{x}$.

The mean function $m(\mathbf{x})$ and the covariance function $k(\mathbf{x}, \mathbf{x}')$ fully specify the GP and for a real process they are defined as

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}\left[f(\mathbf{x})\right] \\ k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}\left[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}) - m(\mathbf{x}))\right] \end{aligned} \tag{2.21}$$

One of the most commonly used covariance functions is the squared exponential (SE) kernel which can be derived from linear regression with infinitely many Gaussian basis functions [16]. An unscaled version of the SE kernel is

$$k_{SE}(\mathbf{x}, \mathbf{x}') = \exp\left(\frac{-(\mathbf{x} - \mathbf{x}')^2}{2l^2}\right) \tag{2.22}$$

where l is the characteristic lengthscale parameter. For inputs that are far from each other with respect to l the covariance tends to zero.

Generally speaking, a function is a map from input space to output space where inputs that are considered close to each other are likely to have similar outputs. The covariance function expresses closeness of two input variables with respect to the characteristics of the function under consideration. For functions with small lengthscales, that is rapid variations over the input space, there is less covariance between two inputs far from each other than for a function with large lengthscales. The covariance function is with other words highly connected to the behaviour or characteristic lengthscales of the function that is modeled. Specifying the functional form of the covariance function implies a prior distribution of possible functions $f(\mathbf{x})$. For instance one could use a periodic covariance function to incorporate periodic behavior or alter the differentiability of the kernel to modify smoothness. Fig. 2.5 shows five random functions, drawn from Gaussian processes defined with the SE kernel in Eq. (2.22), for two different lengthscales together with a heatmap of corresponding covariance matrices $\mathbf{K}$. One can see that for the smaller lengthscale $l = 0.08$, the drawn functions exhibit a lot of variation in the output value over an interval in x, e.g. between 0.25 and 0.50, indicating that there is no correlation between $f(0.25)$ and $f(0.50)$. Indeed the covariance matrix for $l = 0.08$ is seen to be zero for $x = 0.25$, $x' = 0.50$. This is in contrast to the larger lengthscale $l = 0.4$

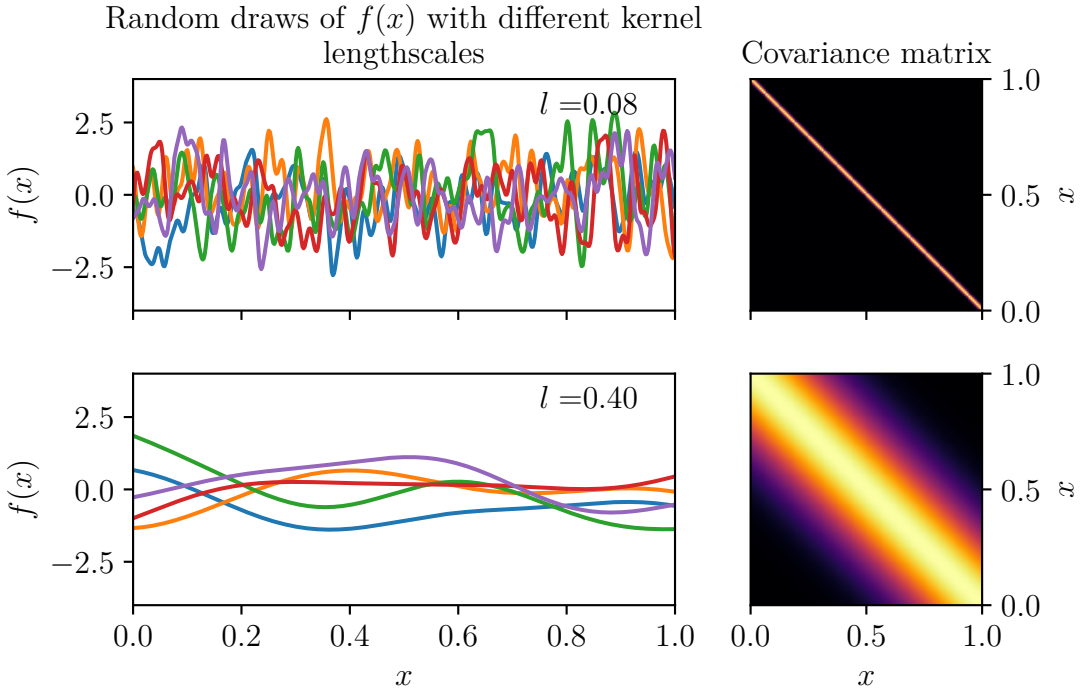

Figure 2.5: Five random draws from a Gaussian process with the square exponential kernel as covariance function (see Eq. (2.22)) and zero mean, above with lengthscale 0.08 and below with lengthscale 0.4. To the right is the heatmap of the covariance matrix.

where the functions do not vary that much on the same interval yielding somewhat similar output values for $f(0.25)$ and $f(0.50)$. The covariance matrix is also not zero for $x = 0.25$, $x' = 0.50$.

2.2.2.1 The Matérn covariance function

As discussed above, characteristics of the covariance function are connected to the characteristics of the function that is modeled. The SE kernel function is infinitely differentiable which means that a GP with this kernel is very smooth. As many physical processes are not that smooth, the *Matérn class* of kernel functions is a common choice suitable for modeling physical systems [16]. The general form for a Matérn kernel is

$$k_{Matern}(r) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} r}{l} \right)^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu} r}{l} \right), \tag{2.23}$$

where $r = |(\mathbf{x} - \mathbf{x}')|$, ν and l are positive (hyper)parameters and K_{ν} is a modified Bessel function. The smoothness of the Matérn kernel increases with ν and for $\nu \to \infty$ it reduces to the ordinary SE kernel [16]. For half integer values of ν the

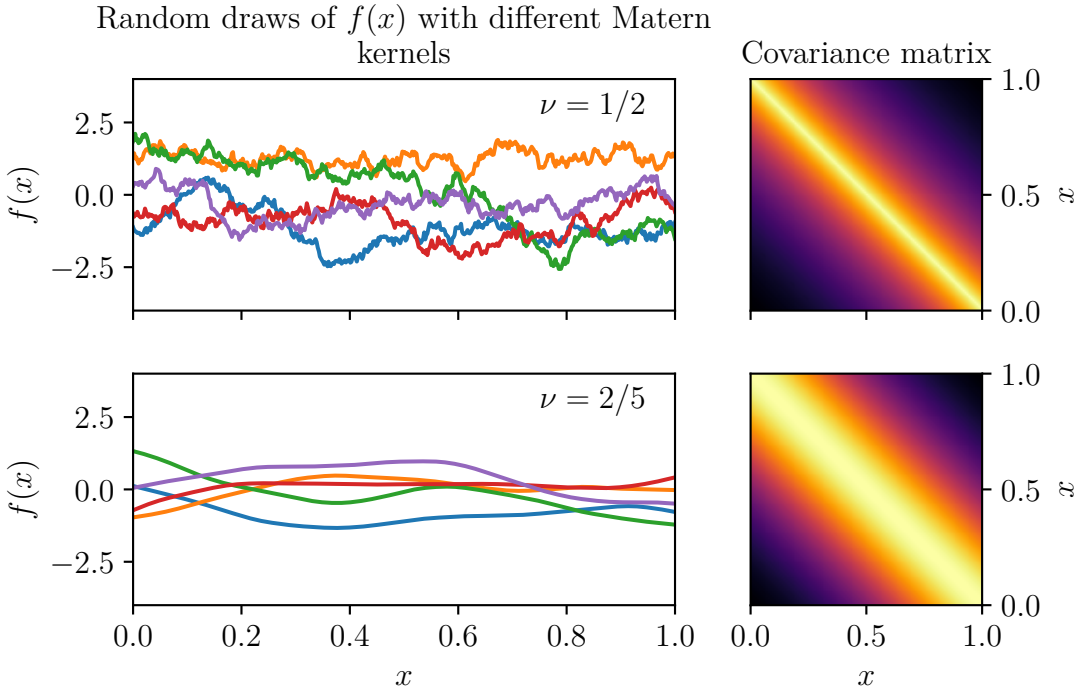

Figure 2.6: Five random draws from a Gaussian process with different Matérn kernels as covariance function and zero mean and lengthscale 0.4. To the right is the heatmap of the covariance matrix.

kernel takes the form

$$k_{Matern}(\mathbf{x}, \mathbf{x}') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\rho\right)^{\nu} K_{\nu} \left(\sqrt{2\nu}\rho\right), \tag{2.24}$$

where $\rho = (\mathbf{x} - \mathbf{x}')^T \Theta^{-1} (\mathbf{x} - \mathbf{x}')$ is the distance between $\mathbf{x}$ and $\mathbf{x}'$ scaled by the lengthcale parameters $\Theta = \text{diag}([l_1, .., l_d])$. In Fig. 2.6, 5 random functions drawn from Gaussian processes defined with the Matérn kernel in Eq. (2.23) with two different smoothness parameters ν, are seen together with a heatmap of the corresponding covariance matrix $\mathbf{K}$. The lengthscale was set to $l = 0.4$. For $\nu = 1/2$ the functions are quite rough. Already for $\nu = 2/5$ the functions are much smoother and similar to the SE kernel with the same lengthscale seen in the lower part of Fig. 2.5.
A covariance function like the SE and the Matérn kernels have the pleasant property of automatic relevance detection (ARD). This can be seen from

$$\rho = (\mathbf{x} - \mathbf{x}')^T \Theta^{-1} (\mathbf{x} - \mathbf{x}') = \sum_{i=1}^{d} \frac{(x_i - x_i')^2}{l_i^2}, \tag{2.25}$$

since a very large lengthscale in a certain dimension would result in the covariance being in principle independent of that dimension and the inference would thus be practically unaffected by this parameter.

The lengthscales l and smoothness parameter ν are referred to as hyperparameters since they parametrize the prior distributions of $f(\mathbf{x})$ rather than the model itself.

These are parameters that are not explicitly interesting but, as discussed above, imply certain properties of the random functions that are interesting for modeling. The noise variance introduced in the next section is also a hyperparameter since it parametrizes the prior distribution of the measurement error ε. Further on, the task for Gaussian process regression is to learn the hyperparameters from data to improve the knowledge about $f(\mathbf{x})$ and ε.

2.2.3 Multivariate normal as Likelihood

The likelihood of measured data (i.e. observations) $\mathbf{y} = \{y_i\}_1^n$, is the probability for measuring $y_1 \wedge y_2 ... \wedge y_n$ given the model $\mathcal{M}$, the target function f and the measurement error ε, that is the probability $P(y_1, ..., y_n | f(\mathbf{x}), \varepsilon)$. Let $\mathbf{y} = [y_1, ..., y_n]^T$ be the vector of measured target function values and $\mathbf{e} = [\varepsilon_1, ..., \varepsilon_n]^T$ the vector of corresponding measurement errors, then we have

$$\mathbf{y} = \mathbf{f} + \mathbf{e}, \tag{2.26}$$

where $\mathbf{f}$ and $\mathbf{e}$ are vectors of random variables. $\mathbf{f}$ is a GP as discussed in Sec. 2.2.2 and is thus distributes as $\mathbf{f} \sim \mathcal{N}(\mathbf{m}, \mathbf{K})$. Assuming Gaussian noise, $\mathbf{e}$ will also be distributed as a multivariate normal $\mathbf{e} \sim \mathcal{N}(\mathbf{m}_\varepsilon, \Sigma_\varepsilon)$ with covariance matrix Σ_ε. We thus have a sum of two multivariate normal distributions yielding a multivariate normal distribution for $\mathbf{y}$ with mean vector $\mathbf{m} + \mathbf{m}_\varepsilon$ and covariance matrix $\mathbf{K} + \Sigma_\varepsilon$, provided $\mathbf{f}$ and $\mathbf{e}$ are independent. That is $\mathbf{y} \sim \mathcal{N}(\mathbf{m} + \mathbf{m}_\varepsilon, \mathbf{K} + \Sigma_\varepsilon)$ and thus the likelihood is a multivariate normal distribution. Commonly the term $\mathbf{m}_\varepsilon = 0$ by the assumption that the noise has a zero mean.

Depending on further assumptions for the noise, the covariance matrix Σ_ε for the random vector (i.e. a collection of random variables) $\mathbf{e}$, will have different characteristics. For instance assuming white noise implies that $\{\varepsilon_i\}_{i=1}^n$ are uncorrelated yielding a diagonal covariance matrix with the finite variances $\sigma_{\varepsilon,i}$ of each ε_i on the diagonal. Another common noise model is to assume $\{\varepsilon_i\}_{i=1}^n$ to be independent identically distributed (i.i.d.) which also yields a diagonal covariance matrix but with the same variance σ_ε for all ε_i implying homoscedasticity. Other types of noise, with correlations, can also be expressed via Σ_ε. The variances for the measurement errors are generally unknown and can be treated as hyperparameters for which inference can be done. If however there are possibilities to make good point estimates of the variances or they are well known for some reason one can just insert that into Σ_ε. modeling heteroscedastic noise could be done by considering the variance as another Gaussian process.

2.2.4 Maximize marginal likelihood (model selection)

To find suitable estimates for the hyperparameters, θ, that parametrize the prior probability distributions for f and ε, e.g. lengthscales and noise variance, it is common to look at the marginal likelihood of Eq. (2.17). The marginal likelihood (or evidence) expresses the probability distribution over all possible data sets $\mathbf{y}$ given

a model (here model refers to a data generating process including prior knowledge about the system). That is, the probability that randomly selected functions f and ε from the specified model class (given hyperparameters θ) would generate data $\mathbf{y}$, i.e. $P(\mathbf{y}|\theta)$. From integration of the posterior in (2.30)

$$\int P(f,\varepsilon|\mathbf{y})df d\varepsilon = \frac{1}{P(\mathbf{y})}\int P(\mathbf{y}|f,\varepsilon)P(f,\varepsilon)df d\varepsilon = 1 \tag{2.27}$$

the marginal likelihood can be expressed in terms of an integal over the likelihood and the priors according to

$$P(\mathbf{y}|\theta) = \int P(\mathbf{y}|f,\varepsilon)P(f,\varepsilon|\theta)df d\varepsilon. \tag{2.28}$$

The marginal likelihood is a function of θ[1]. Maximization of the marginal likelihood with respect to θ, i.e. finding the model that most likely predicts the data set, is thus often used to select suitable hyperparameters for the given data.

The full Bayesian way to handle hyperparameters θ is to account for the prior knowledge about the hyperparameters via *hyperpriors*. This is done by extending the posterior in Eq. (2.30) to also include θ according to

$$P(f,\varepsilon,\theta|\mathbf{y}) = \frac{P(\mathbf{y}|f,\varepsilon)P(f|\theta)P(\varepsilon|\theta)P(\theta)}{P(\mathbf{y})}, \tag{2.29}$$

where $P(\theta)$ is the hyperprior. Eq. (2.29) is then marginalized over θ to yield the joint posterior for f and ε according to

$$P(f,\varepsilon|\mathbf{y}) = \int P(f,\varepsilon,\theta|\mathbf{y})d\theta. \tag{2.30}$$

This approach is however computationally quite expensive and a *maximum a posteriori* (MAP) estimate is a reliable option. This amounts to maximizing a posterior distribution of θ

$$P(\theta|\mathbf{y}) \propto P(\mathbf{y}|\theta)P(\theta), \tag{2.31}$$

where $P(\mathbf{y}|\theta)$ is the marginal likelihood in Eq. (2.28).

2.2.5 Making predictions from posterior distribution

Given data and a model after hyperparameter fitting, the posterior in Eq. (2.30) can be calculated (or sampled). In the case of a multivariate normal likelihood as discussed in Sec. 2.2.3 This posterior represents our current knowledge about f and ε and this inferred knowledge is used when we want to make predictions about new data points y_p based on known data $\mathbf{y}$. We can express the probability of y_p given $\mathbf{y}$ as a marginalization of the joint probability distribution $P(y_p, f, \varepsilon|\mathbf{y})$ according to

$$P(y_p|\mathbf{y}) = \int P(y_p, f, \varepsilon|\mathbf{y})df d\varepsilon = \int P(y_p|f,\varepsilon)P(f,\varepsilon|\mathbf{y})df d\varepsilon, \tag{2.32}$$

[1]By the assumption of multivariate normal as likelihood $P(\mathbf{y}|f,\epsilon) = \mathcal{N}(y|\mathbf{m}, \mathbf{K} + \Sigma_\varepsilon)$. In the integral one can set $\mathbf{m}$ to 0 by the variable change $f^* = f - y$.

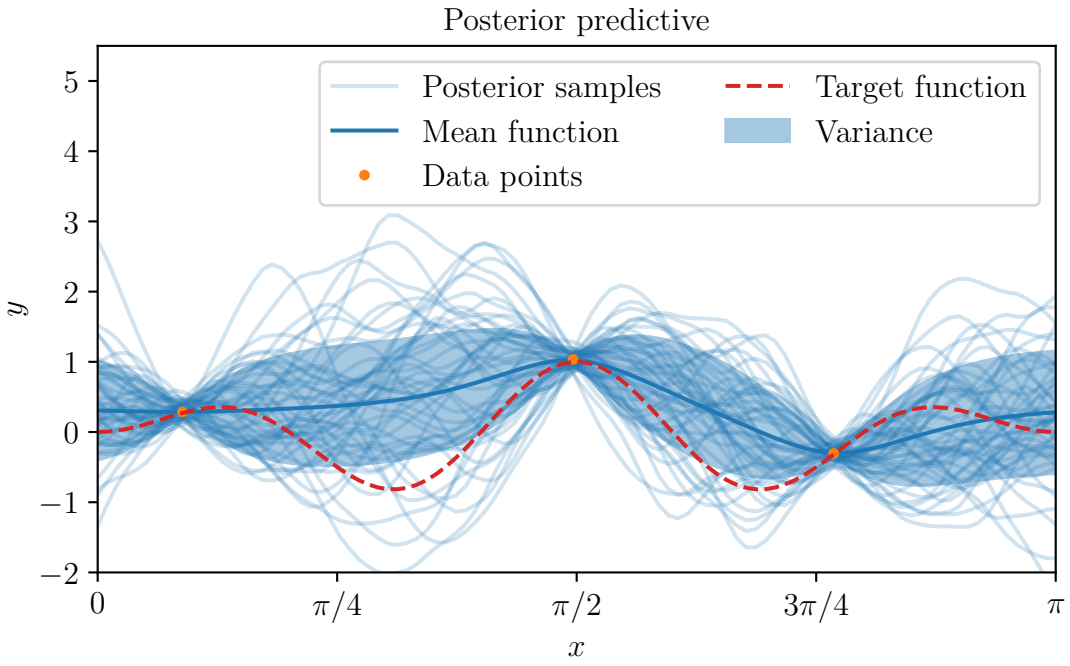

Figure 2.7: Posterior predictive distribution after conditioning on three data points y, these are shown with an orange dot. The blue shadowed lines are 50 draws from the distribution. The shadowed area is the two standard deviations and the blue line is the mean of the distribution. The red line is the underlying target function. A normally distributed noise term with $\sigma = 0.1$ has been added to the target function. Note that standard deviation at the location of the data points is not zero. To compensate for the noise in the target function an equally large standard deviation has been set in the covariance matrix of the noise in the likelihood (see Sec. 2.2.3).

where the chain rule for probability distributions have been used in the last equality and $P(y_p|f,\varepsilon)$ is a multivariate normal likelihood as described is Sec. 2.2.3. Similarly, for predicting new function values $f_p = f(\mathbf{x}_p)$ we have the posterior predictive distribution

$$P(f_p|\mathbf{y}) = \int P(f_p, f|\mathbf{y})df = \int P(f_p|f)P(f|\mathbf{y})df, \tag{2.33}$$

where $P(f|\mathbf{y})$ is the posterior in Eq. (2.30) marginalized over ε. From the assumption that $f \sim \mathcal{GP}$ the conditional distribution $P(f_p|f)$ is also a GP due to the marginalization property of Gaussian processes. Furthermore, for a multivariate normal likelihood, the posterior will also be a multivariate normal and the integral in Eq. (2.33) yields jet another multivariate normal distribution [16]. Eq. (2.33) is referred to as the *posterior predictive distribution* for the target function f and is extensively used in the Bayesian optimization algorithm discussed in the following section. In Fig. 2.7 samples of the posterior for a Gaussian process conditioned on three data points are shown.

2.2.6 Bayesian Optimization

The idea behind Bayesian optimization (BO) is to model the target function with Gaussian process regression and in an iterative manner choose to evaluate points closer to the global maximum. An *acquisition function* chooses which point to evaluate next. The BO algorithm is initialized with a set of randomly sampled data points and the posterior predictive distribution in Eq. (2.33) is conditioned on these points. This distribution represents our model of the target function Eq. (2.15). Note that the conditioning of new data includes an optimization of the marginal likelihood w.r.t hyperparameters according to Sec. 2.2.4. An acquisition function $a(\mathbf{x})$ is then calculated based on the posterior predictive. Optimization of $a(\mathbf{x})$ is performed with a standard optimization algorithm yielding a new input point to sample the target function at. The model is then updated with the new data which entails updating the posterior predictive and maximizing the marginal likelihood for optimization of the hyperparameters. In essence the BO algorithm consists of the following 4 steps

1. Evaluate target function at point $\mathbf{x}$ to get y
2. Maximize the marginal likelihood conditioned on the new data points $[y, \mathbf{x}]$ to get optimal hyperparameters
3. Update the posterior distribution with the new hyperparameters
4. Maximize the acquisition function based on the posterior distribution to get a new evaluation point $\mathbf{x}$

There are many ways to formulate the acquisition function, generally it should express some trade off between favouring new points where the model predicts high function values (*exploitation*), that is $\mathbb{E}[f_p]$ is high, and points where the uncertainty of the model is large (*exploration*), that is where the model captures the true function badly [17]. The specific acquisition functions, *Expected Improvement* and *Noisy Expected Improvement*, explored in this thesis are described in Sec. 2.2.7.

In Fig. 2.8 the evolution of BO with Expected Improvement as acquisition function is seen. As more data is added the maximum of the acquisition function $a(x)$ changes and the mean function $\mu(x)$ adapts to the true objective function. In the first graph the model only has two data points, the acquisition function chooses to evaluate the function at $x \approx 1$. The marginal likelihood is then optimized with respect to this new data point and a new acquisition function is calculated. After one more iteration the posterior predictive looks like the second graph in Fig. 2.8. The suggested point to evaluate is now pretty close to the actual maximum. After 3 more BO iterations the acquisition function deems that the current maximum is accurately described and moves on to explore features of the function in other locations. This can be seen in the last graph of Fig. 2.8 where the acquisition function is higher in areas unknown to the model. Note that the amplitude of the standard deviation in areas unknown to the model is varying a lot depending on the amount of data points available. This is simply a consequence of the optimization of hyperparameters θ (see Sec. 2.2.4).

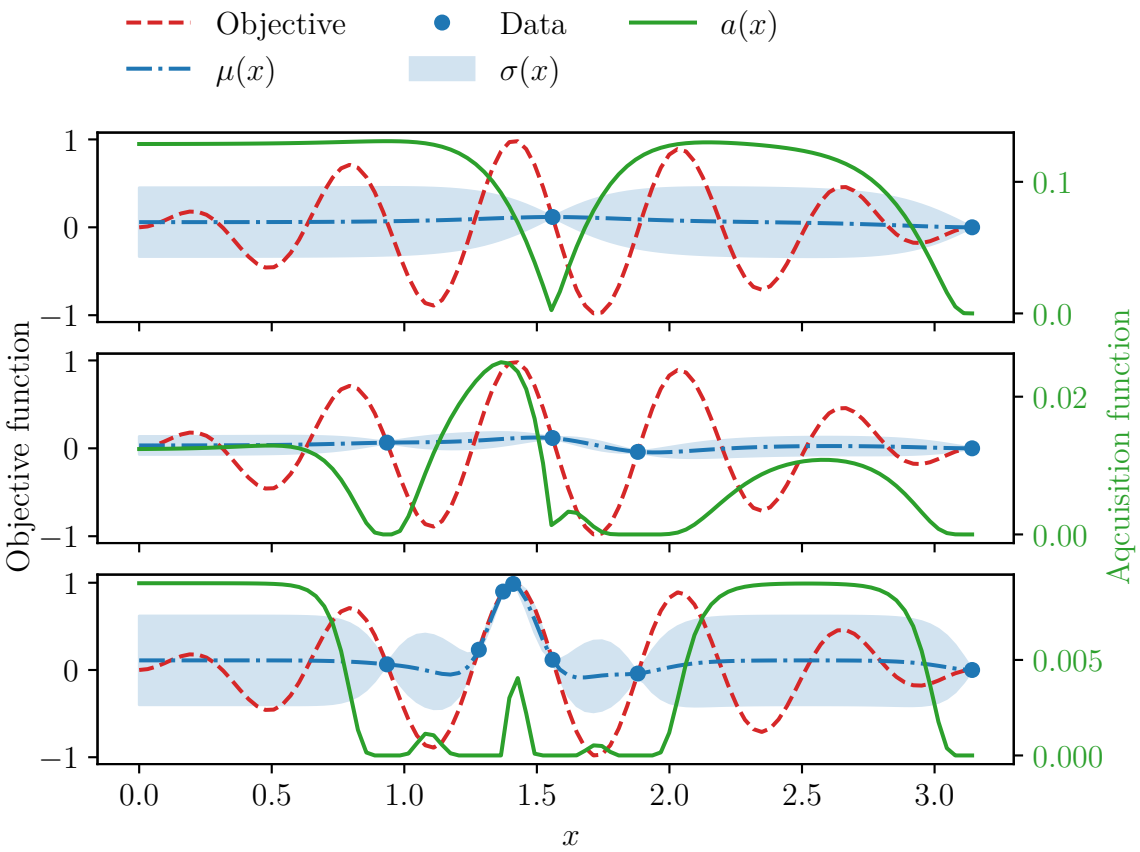

Figure 2.8: Evolution of BO on the objective function $f = x\sin(x10)$ showed with a red dashed line. The blue dots are sampled data from the objective function, the blue dashed-dotted line is the mean of the posterior predictive and the blue shadowed area is the 68% confidence interval of the posterior predictive i.e. 2σ. The green line is the acquisition function, which decides which point that should be sampled in the next iteration.

2.2.7 Improvement based acquisition functions

There are numerous acquisition functions that one might come up with, we have decided to focus on a group of acquisition functions that is trying to quantify the expected improvement of measuring a specific point, these are called improvement-based acquisition functions.
The virtue of improvement-based acquisition functions is that they adaptively balances exploration and exploitation [17]. Improvement based acquisition functions rely on quantifying the expected increase of the best value [17]. In other words they try to give an estimate of how much the current best value suggested by the BO would increase by doing a measurement at a specific coordinate. The simplest improvement-based acquisition function is Expected Improvement (EI). It can be defined as

$$EI[\mathbf{x}_p] = \mathbb{E}\left[(f(\mathbf{x}_p) - y_{max})^+\right] \tag{2.34}$$

where y_{max} is the largest target function value measured so far, $f(\mathbf{x}_p)$ a random variable with a distribution given by the posterior and $a^+ := \max(0, a)$ [17]. Expected improvement is Bayes-optimal on two conditions: (1) the target function is noise free, (2) the best target function value y_{max} is restricted to previously sampled points [17].

If the target function is not noise free one can modify EI to account for this noise, one example of this is Noisy Expected Improvement (NEI). The problem with EI in a noisy environment is that the value y_{max} might not be a good approximation of the (noise free) target function in that point. Therefore NEI uses the posterior in the previously measured points and calculates an expectation value of the difference between the maximal previously measured point and the value at $\mathbf{x}$ as

$$\mathrm{NEI}(\mathbf{x}_p) = \mathbb{E}\left[\left(f(\mathbf{x}_p) - \max_{\mathbf{z}\in\mathbb{D}} f(\mathbf{z})\right)^+\right] \tag{2.35}$$

with $\mathbb{D} = \{\mathbf{x}^{(i)}\}_{i=1}^n$ where $\mathbf{x}^{(i)}$ is the ith measured point [18]. Note that here also $\max_{\mathbf{z}\in\mathbb{D}} f(\mathbf{z})$ is a random variable.

For finding the maximum of the acquisition function it is possible to calculate an analytical expression for both EI and NEI [17][19], for derivation of an analytical expression for EI see App. A.5. One can also treat the problem deterministically meaning that one can approximate the acquisition function at $\mathbf{x}$ by MC sampling of $f(\mathbf{x})$ from the posterior distribution [20]. For example EI is then approximated by

$$\widetilde{EI}[\mathbf{x}_p] = \frac{1}{n}\sum_{i=0}^{n}(f(\mathbf{x}_p)_i - y_{max})^+ \tag{2.36}$$

where $\{f(\mathbf{x}_p)_i\}_{i=0}^n$ is sampled from the posterior distribution of $f(\mathbf{x}_p)$ [20].

3

Methods

The LUX beamline holds a unique monitoring system enabling on-the-fly precise control of some of the input parameters connected to the driving laser and the gas composition in the plasma target. As mentioned in Sec. 2.1.1.3 these are the longitudinal focus position of the driver laser z_{foc}, the energy of the driver laser E_{laser}, and the gas flows of hydrogen and nitrogen in the three inlets. The target function in Eq. (2.15) was in this project optimized with respect to these quantities.

Since the access to the actual experiment is limited, due to maintenance of the driver laser and limited material resources, one has to resort to computational tools for simulation of the experiment and investigation of different setups of the BO algorithm. One such tool is Particle-In-Cell (PIC) simulations which has the benefit of being a well known and extensively used tool for modeling laser-plasma interactions [21]. To compare the performance of BO with EI and NEI in a noisy environment optimization was performed using PIC simulated data to evaluate the target function with noise added to the output. The details of the PIC simulations are described in Sec. 3.1 and the BO algorithms used is specified in Sec. 3.2.

In a 3D setting, one PIC simulation with (for our purposes) good resolution took about 16 minutes using computational resources available at DESY. For BO, typically employing at least 100 iterations for a multidimensional input space, this means that a single run of optimization would take about 20 hours. Note that for other common optimization algorithms that need thousands of iterations this would not be feasible at all. Furthermore, if different strategies for sampling of each set point are to be investigated, e.g taking the mean of several points around each set point, this drastically increases the number of PIC simulations that would be needed. This motivated an attempt to further approximate the multidimensional surface corresponding to Eq. (2.15) and then perform trial optimization runs on this model at the time scale of less than an hour. In this thesis Gaussian processes were exploited, once again, together with PIC simulated data to make such model. Depending on the dimensionality of the modeled output space these models are referred to as *Single-output models* or *Multi-output models* and the details of the training and evaluation is presented in Sec. 4.2.2. For realistic modeling of the noise in the experiment, a normally distributed random value was added to the input parameters and the output of the models. The specifics about how noise was modeled can be read in Sec. 3.4.

3.1 Particle in cell simulations

Particle-in-Cell codes can describe the detailed motions of a large number of charged particles in electromagnetic fields. They are based on the idea that it is possible to describe a plasma accurately by considering an ensemble of particles instead of the position of separate particles [22]. The theory is thus similar to the framework that was developed in previous sections covering the dynamics of plasma. It is assumed that the charged particles generating the fields only interact with each other through the self-sustained fields **E** and **B** [22], there are no particle-particle interactions. The fields are calculated on a grid giving the code its name: Particle-In-Cell, but the trajectories are calculated in phase space. To make this feasible the particles are bunched into macro particles [22]. The forces exerted on the macro-particles are the same as the forces exerted on the separate particles since they have the same mass-to-charge ratio [22].

In this work a PIC code called Fourier-Bessel Particle-In-Cell (FBPIC) was used. This code was originally developed by Remi Lehe at Berkeley Lab, and Manuel Kirchen at CFEL, Hamburg University. A special feature of the FBPIC code is that it decomposes the electromagnetic fields into azimuthal modes to utilize the close to cylindrically symmetric nature of wakefields. This allows the use of a set of radial 2D grids, one for each mode, instead of the computationally expensive 3D cartesian grid. For the simulations in this thesis three of these modes were used. The two first modes are sufficient to accurately capture an idealized wakefield, independent of the azimuthal angle θ, and a linearly polarized laser field, varying proportional to $\cos(\theta)$. Higher modes capture further deviations from the cylindrical symmetry. A linearly polarized laser with a flattened transverse Gaussian profile was used with a spot size in the focal plane of about 16 µm and pulse duration of about 29 fs. The energy of the laser, E_{laser}, and the focus position, z_{foc}, were parameters subjected to optimization as discussed in previous sections.

The laser was propagated through a cylindrically symmetric preionized plasma (since outer shell electrons are ionized immediately when the laser enters the gas) with plasma density (i.e. electron density) determined by the longitudinal density profile of the hydrogen and nitrogen gases seen in Fig. 3.1. The gas densities were parametrized by the plasma density, n_0, at the plateau where the acceleration occurs and by the concentration of nitrogen, c_{N_2}, resulting in higher or lower plasma density peak at the beginning of the plasma. The parametrization was chosen based on a fit to fluid dynamic simulations of the gas target made in previous studies [9].

In the lower row of Fig. 3.1, three different configurations of c_{N_2} (mint green) and the z_{foc} (red vertical line) can be seen. The resulting longitudinal phase space of the accelerated electron bunch is shown in the upper row of the figure. It is clear that the parameter configurations have a clear impact on the shape of the electron beam.

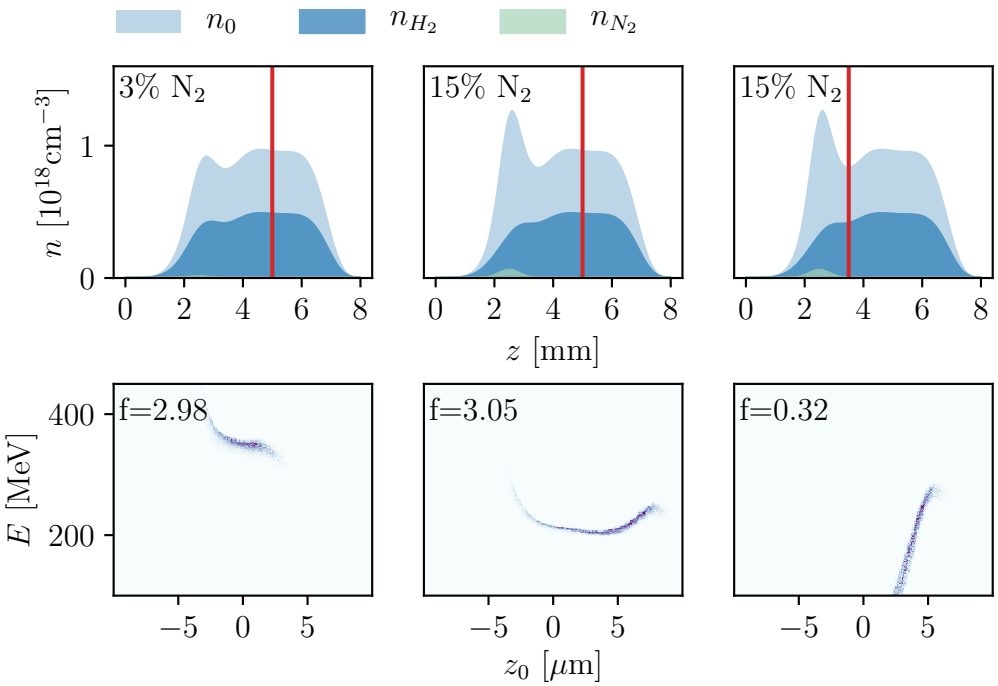

Figure 3.1: The upper figures show the gas and plasma density distributions together with the focus position of the laser. The lower figures show the resulting phase space distributions of the electron bunch for the different parameter configurations.

The median energy $\tilde{E}$ and median absolute deviation ΔE of the accelerated electrons were calculated from macro particles with longitudinal momentum corresponding to energy higher than 25 MeV. The total charge Q of these electrons were registered and the target function value f was calculated from Eq. (2.15). The target function value was set to zero for simulations resulting in beams with a total charge less than $\sim 0.2 \cdot 10^{-7}$ pC since the number of electrons in such beams are too few for the energy statistics to be reliable.

It should be emphasised that the four parameters allowed to vary in the PIC simulations during the optimization does not capture the full complexity of the true experiment. In a real laser-plasma accelerator there are, of course, many more parameters influencing the system, both known parameters that can be controlled to a certain degree like the pulse duration and spot size of the driver laser, but also parameters that are more difficult to characterize like inhomogeneities in the wave front and pressure fluctuations in the plasma.

3.2 Bayesian optimization with *BoTorch*

The implementation of the Bayesian optimization algorithm was made with the python library *BoTorch* specifically designed for Bayesian optimization [20]. It is built upon *PyTorch* enabling efficient tensor calculations on GPUs and uses the GP

models from *GPyTorch*. As acqusition functions BoTorch's *batch Noisy Expected Improvement* (qNEI) and *batch Expected Improvement* (qEI) were chosen. In general it is not possible to find an analytical expression for the maximum of the acquisition functions [20]. However one can treat the problem deterministically meaning that we can approximate the acquisition function at $\mathbf{x}$ by sampling $f(\mathbf{x})$ from the posterior [20]. This is also the computational method behind qEI and qNEI. The standard kernel used in *BoTorch* is the *Matérn kernel* introduced in Sec. 2.2.2.1 with $\nu = 3/2$. After trying different hyperpriors for the hyperparameters (i.e. lengthscale, noise variance and output scale) it was clear that the standard *Gamma* priors which worked well with normalization of the input data was the best choice of priors. For the internal modeling in the BO algorithm the *SingleTaskGP* BoTorch class was used where a multivariate normal likelihood with homoscedastic noise is assumed, as discussed in Sec. 2.2.3. The input data was normalized to the search space interval of each parameter and the output data was standardized. This is a standard procedure and with approximately normally distributed data it ensures that the hyperpriors of lengthscales and output scales are appropriate.

3.3 modeling the target surface with Gaussian process regression

Since high resolution PIC simulations are very costly a model was built using Gaussian process regression to mimic the PIC surface. This was done by preforming PIC simulations at 8514 randomly sampled points from the six-dimensional parameter space with ranges seen in Tab. 3.1. The additional parameters τ and w were introduced to include the effect of variations in pulse duration and spot size of the driver laser in the PIC simulations. After sorting out points with charge lower than 10 pC and energy median lower than 20 MeV, with the motivation that these points were too low to yield realistic data in the energy spread, there were 6182 points left. This set is called the *base set*. When examining this data set one distinctive max was found. To improve the model additional data was sampled in the vicinity of this point. This set is called the *refined set* and it consist of 160 points after filtering out low charge and energy points. Several ways of building and training the Gaussian process were evaluated and in the end two models, one multi-output model and one single-output model, were chosen from their prediction performance. The multi-output model was used for investigation of the propagation of noise in the input parameters and the single-output model was used for evaluating BO.

The single-output model was trained on the six-dimensional input space, consisting of laser energy E_{laser}, focus position z_{foc}, plasma density n_0, nitrogen ratio c_{N_2}, pulse duration and laser spot size, and on the one-dimensional output space consisting of the target function. The multi-output model was trained on the six-dimensional input space and on a three-dimensional output space, consisting of charge, energy median and energy spread. The Gaussian process regression model of the multi-output model can be seen as three separate single-output models since no correlation was induced between the outputs. For convenience we will still refer to it as a

multi-output model. The final models were built using GPyTorch. Building Gaussian process models can also be done using the *SingleTaskGP* class of BoTorch. The Single-output models with fixed noise were built with this class (see Tab. 4.2) since the BoTorch model seemed to perform better or equal for this particular setting. For models with inferred noise levels and in particular multi-output models with a lot of data BoTorch models performed worse. This might be a consequence of the way BoTorch is sending data to the GPU (and how the particular GPU used was structured) resulting in worse performance than similar algorithms built with GPyTorch. For this reason GPyTorch classes were used for building the multi-output model and the single-output models with inferred noise. Further the GPyTorch library is a more low-level library than BoTorch which is built upon GPyTorch. This gives more control of the training process, for example one can generate loss plots which is of great help for evaluation of the training.

To compare performance of different models the data sets were split into 80% training data and 20% validation data. Two different scores were used, the mean square error (MSE) and the R^2 value. The MSE expresses the sum of squared differences between predicted output value and true output value defined as

$$MSE = \frac{1}{n}\sum_{i=0}^{n}(y_p(x_i) - y_i)^2, \tag{3.1}$$

where y_p is the prediction of the modeled function and y_i is the target function. The R^2 value is defined as

$$R^2 = 1 - \frac{\sum_{i=0}^{n}(y_p(x_i) - y_i)^2}{\sum_{i=0}^{n}(y_i - \hat{y})^2}, \tag{3.2}$$

where $\hat{y}$ is the mean value of the sampled data y_i. Thus the R^2-value quantifies the proportion of variance that is explained by the model to the variance of the input data. A R^2 value of 1 means that the prediction and data is perfectly correlated, i.e. the model is a perfect model.

3.4 Noise in LUX experiment

In all experimental setups there are sources of noise. Since the algorithm employed at the LUX beamline uses both the input and the output data of the system to build a model one has to be aware of (1) measurement noise on the input parameters, (2) measurement noise on the output parameters and (3) noise introduced by the system. From previous experiments and in discussions with experts on the system (see [9]) it was possible to make a qualitative guess of the noise in the input parameters of the system which can be seen in Tab. 3.1. Further, a *jitter* was introduced to the laser focus parameter. This parameter fluctuates substantially during experiments but can be measured with a higher accuracy. This was accounted for by adding a random term to the focus that is later passed to the querying function (i.e. jitter). In essence the following is the data generating procedure:

1. Query of function value at $\mathbf{x}$

2. $\mathbf{x}_{jitter} = \mathbf{x} + \epsilon_{jitter}$
3. $\mathbf{x}_{noisy} = \mathbf{x}_{jitter} + \epsilon$
4. $f_{noisy} = f(\mathbf{x}_{noisy})$
5. return f_{noisy}, $\mathbf{x}_{jitter}$.

The function f is either a PIC simulation or surrogate model built from PIC simulations.

Table 3.1: The standard deviation and range of the input parameter noise and the input jitter. For convenice the spot size and pulse length has been scaled by the standard values used in the PIC simulations.

Parameter	Input noise (ϵ)	Jitter (ϵ_{jitter})	Range
E_{laser} [J]	0.0256	0	$[1.28, 3.84]$
z_{foc} [mm]	0.01	0.2	$[0, 7.0]$
c_{N_2} [%]	0.0001	0	$[1, 30]$
n_0 [10^{18}cm^{-3}]	0.00001	0	$[0.5, 1.5]$
w [μm]	0.01	0	$[0.8, 1.2]$
τ [fs]	0.01	0	$[0.8, 1.2]$

4
Results

This chapter starts with presenting the results from BO optimizations with PIC simulations as target function. Two different acquisition functions: qNEI and qEI were evaluated and of the two qNEI showed a better performance. The results are shown and discussed in Sec. 4.1. Next the results from building a target function by Gaussian process regression models is discussed. This was done by randomly sampled PIC data over the space in Tab. 3.1 and with the aim of quick evaluation of different BO algorithms. To evaluate the propagation of input noise in the output parameters charge (Q), energy median ($\tilde{E}$) and energy spread (ΔE) a multi-output model was built and trained. In Sec. 4.2.1 distributions generated by this model are compared with earlier experimental results from the system and characteristics of the distributions are tied to physical dynamics of the system through the PIC simulations in Fig. 4.4. For evaluation of different BO algorithms a model, trained directly on the target function values generated by PIC simulations, was built. The results of this work can be seen in Sec. 4.2.2. Using the single-output model different BO algorithms were evaluated, the result can be seen in Sec. 4.3. In this section sampling strategies are compared - taking the mean of several measurements versus only using one measurement at each set point. Not too surprisingly the former strategy was superior.

4.1 Bayesian optimization on PIC simulations

For comparing BO with the acquisition functions qNEI and qEI the algorithms were executed on the PIC simulated target function as described in Sec. 4.2.2. In Fig. 4.1 the predicted maximum from the BO algorithm, i.e. the maximum of the mean function $\mu(\mathbf{x}_{max})$, has been plotted as a function of BO iterations. It is clear that qNEI finds some higher maximum that qEI does not. In Fig. 4.2 the convergence of the parameter values corresponding to the maximum mean function value ($\mathbf{x}_{max}$) is shown. One can see that qNEI has a slightly more precise and faster convergence on the parameters E_{laser} and z_{foc}. These parameters are also the parameters where one can see the largest change in the longitudinal phase space of the accelerated electrons when varying around one point, this can be seen in Fig. 4.4. In the n_0 and c_{N_2} dimension there are no clear convergence. This suggests that the surface is either very flat or very detailed so that one clear maximum is hard to distinguish in these dimensions.

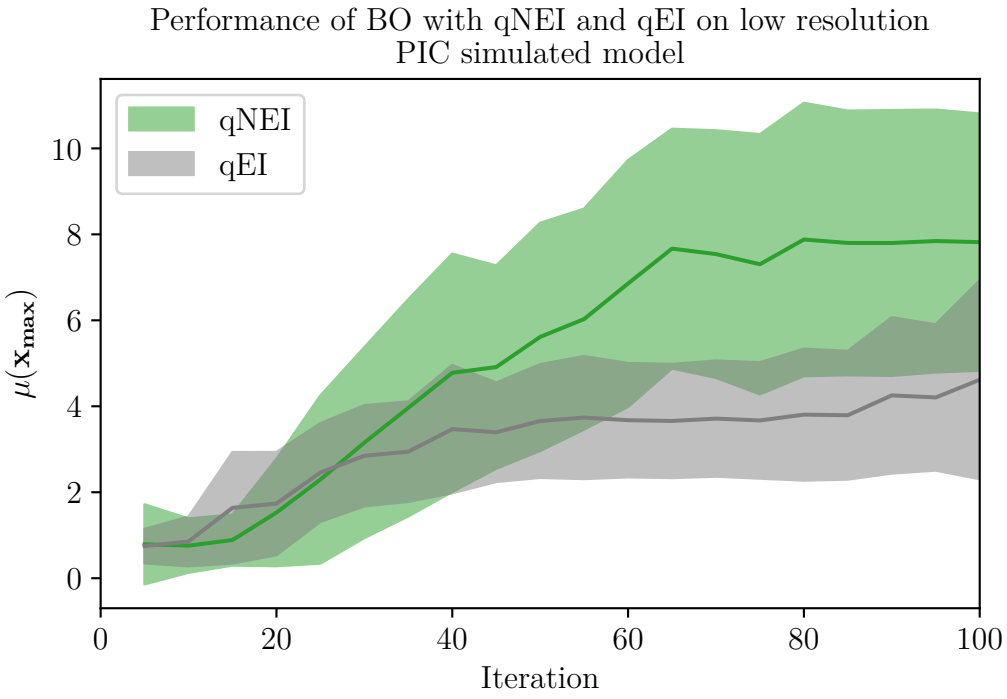

Figure 4.1: Bayesian optimization on a PIC simulated target function using qNEI (green data set) and qEI (gray data set) acquisition function. Five initial points were used followed by 95 BO iterations.

The BO was executed on the entire search space which can be seen in Tab. 3.1. A smaller search space could be considered to improve the convergence, especially for E_{laser} and z_{foc} where the algorithms converge to a smaller area compared to c_{N_2} and n_0. Gaussian noise was added to the output with standard deviation of 0.2 which corresponds to approximately 1% of the maximum that was found from optimization with qNEI. Additionally there are some inherent fluctuations in the PIC simulations itself due to the stochastic projections of electrons into macro particles. At the found optimum from the optimization with qNEI, 100 PIC simulations were executed to give a quantification of these fluctuations. The standard deviation was found to be 0.44 which is quite high compared to the added Gaussian noise.

From the separate runs of Fig. 4.1 the final maximal mean function value over all runs was extracted and corresponding input parameters $\mathbf{x}_{max}$. At and around this point further PIC simulations were made to validate the existence of a maximum at this point. Fig. 4.3 shows the longitudinal phase space of the beam at this point and the plasma density profile with z_{foc} marked with a red vertical line. As can be seen, the beam is flat in phase space, which entails a low energy spread.

In Fig. 4.4 the effect of varying different parameters around this maximum is shown. For each row in the figure, one of the input parameters is varied with 5% of the total search space for that variable around the optimum $\mathbf{x}_{max}$. Note that the figures in the middle column are all at the optimal input but the target function value f varies due to the inherent fluctuations of the PIC simulation. As one can see z_{foc} has a

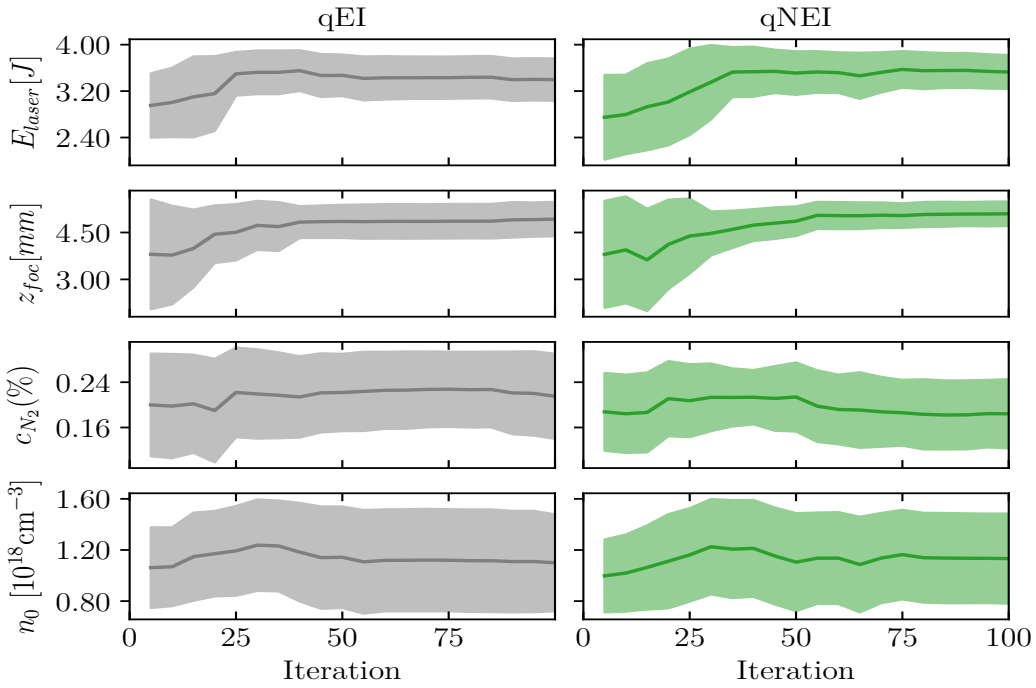

Figure 4.2: The parameter evolution during Bayesian optimization for PIC simulations using the qEI (left) and the qNEI (right) acquisition function. Five initial points were used followed by 95 BO iterations.

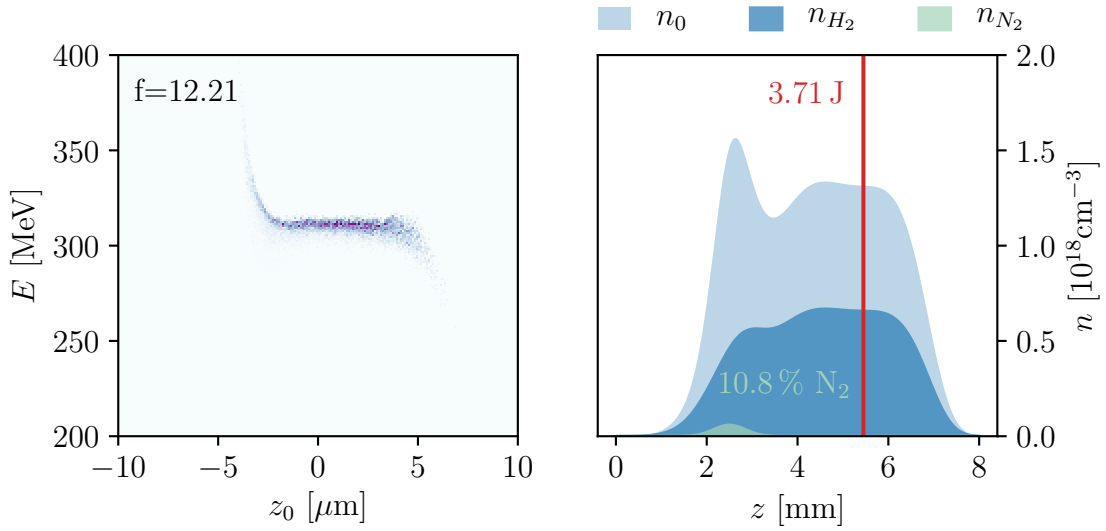

Figure 4.3: Longitudinal phase space of accelerated electron bunch (left) for optimized input parameters (right) using qNEI acquisition function. The vertical red line marks the position of the laser focus z_{foc}. The optimal parameters were $E_{laser} = 3.71$ J, $z_{foc} = 5.45$ mm, $c_{N_2} = 10.8$ % and $n_0 = 1.09 \cdot 10^{18}$ cm^{-3}.

strong effect on both the median energy and the energy spread. This is because when changing the focus position the laser intensity changes at the position where the nitrogen is located, resulting in more or less ionization and there by the amount of injected charge. Furthermore, if the injection is not at the position of the density down ramp then the beam loading process will be affected and in extension the energy spread and median energy. In [9] this effect was coupled with the associated charge of the bunch. For a high charged beam the slope of the accelerating field becomes locally inverted creating a positive correlation in the energy spectra [9]. For a low charged beam however the effect of beam loading, i.e. the flattening of the accelerating field, got weaker which pushes the end of the beam into higher energies and creates a negative correlation. When varying the laser energy one can see a small convex curvature of the phase space for lower values and a concave curvature for higher. The nitrogen concentration C_{N_2} leaves the phase space distribution almost unchanged but a small decline is visible in median energy as the concentration increases. The charge is affected since C_{N_2} governs the nitrogen doping of the gas and thereby the amount of electrons available for ionization injection, however the increase of nitrogen concentration also leads to an increase in plasma density, n_0, affecting the accelerating field. Like for the nitrogen concentration the phase space distribution is barely changed by varying the plasma density, one can however see a small tilt of the phase space and an increase in target function value. The plasma density foremost changes the strength of the accelerating field, i.e. the creation of the bubble structure in the blowout regime. This will naturally affect the beam loading which in turn affects the energy spectra. One should however note that for variations of both plasma density and the nitrogen concentrations the change in target values are on the order of the inherent fluctuations in the PIC simulations making any firm conclusions impracticable.

To the left in Fig. 4.5 the inferred noise level σ_ε from the BO with qEI and qNEI is shown. The final noise level for each run is correlated to the corresponding maximum mean function value $\mu(\mathbf{x}_{max})$, sorted by the maximum mean value. σ_ε is quite similar for BO with qNEI compared to BO with qEI. For qEI there is a correspondence between high σ_ε and low $\mu(\mathbf{x}_{max})$. A high σ_ε implies that more data points are regarded as noise and generally larger lengthscales are expected since the mean function is not obliged to fit every data point. To the right in Fig. 4.5 one can see that it is indeed so that qEI has a tendency towards larger lengthscales for the z_{foc} parameter compared to qNEI. The figure shows final lengthscales $l_{z_{foc}}$ as a function of $\mu(\mathbf{x}_{max})$ for each run and a clear relation between high $\mu(\mathbf{x}_{max})$ and low $l_{z_{foc}}$. A majority of the qNEI runs have $l_{z_{foc}}$ less than 0.05 compared to qEI.

For validation of the results presented above an additional comparison was made between qNEI and qEI with higher resolution settings of the PIC simulations. This took considerably longer time and therefore only 16 runs for each acquisition function were made. The results of these runs can be seen in Fig. 4.6 where all the individual runs have been plotted. 4 out of 16 runs with the qNEI acquisition function found a higher maximum, these runs are marked as green lines.

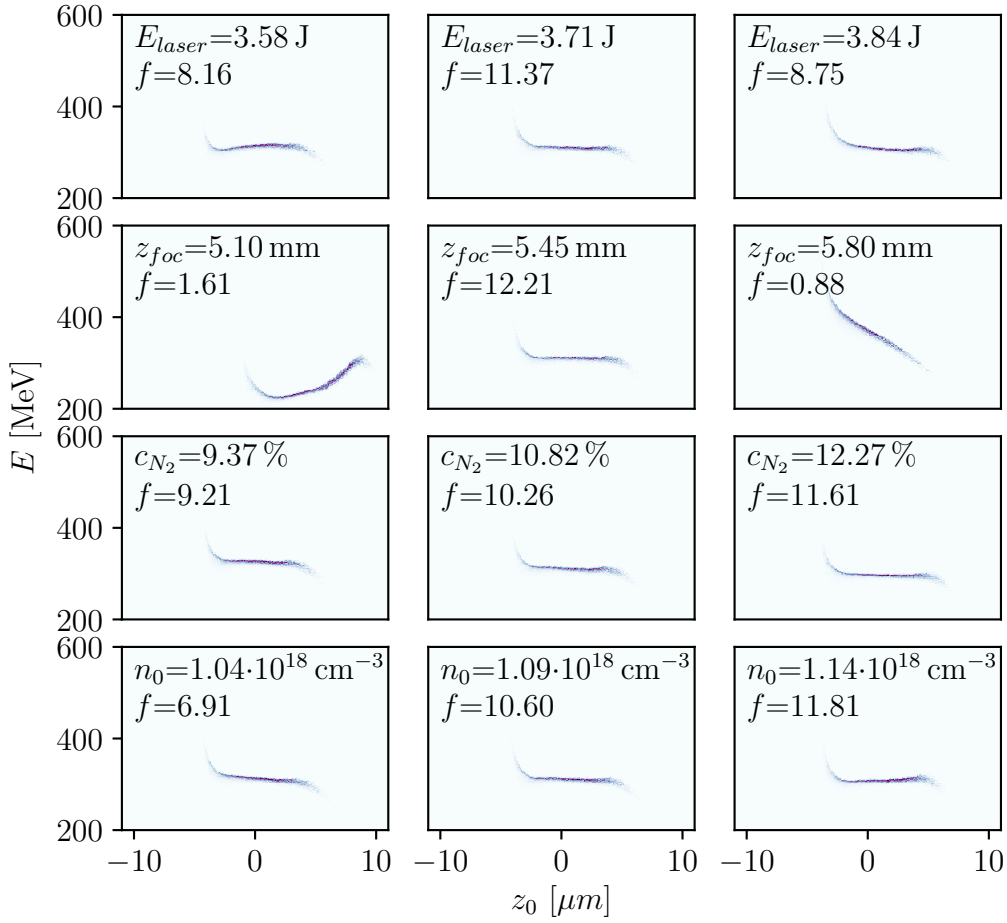

Figure 4.4: Longitudinal phase space for PIC simulated electron beams with different input parameters around the found optimal set-point. For each row one of the input parameters is varied with 5% of the total search space for that variable around the optimum with E_{laser} = 3.71 J, z_{foc} = 5.45 mm, c_{N_2} = 10.8 % and $n_0 = 1.09 \cdot 10^{18}$ cm^{-3}. Note that the beams in the middle column have the same input parameters but the target function value f varies due to the inherent fluctuations of the PIC simulations.

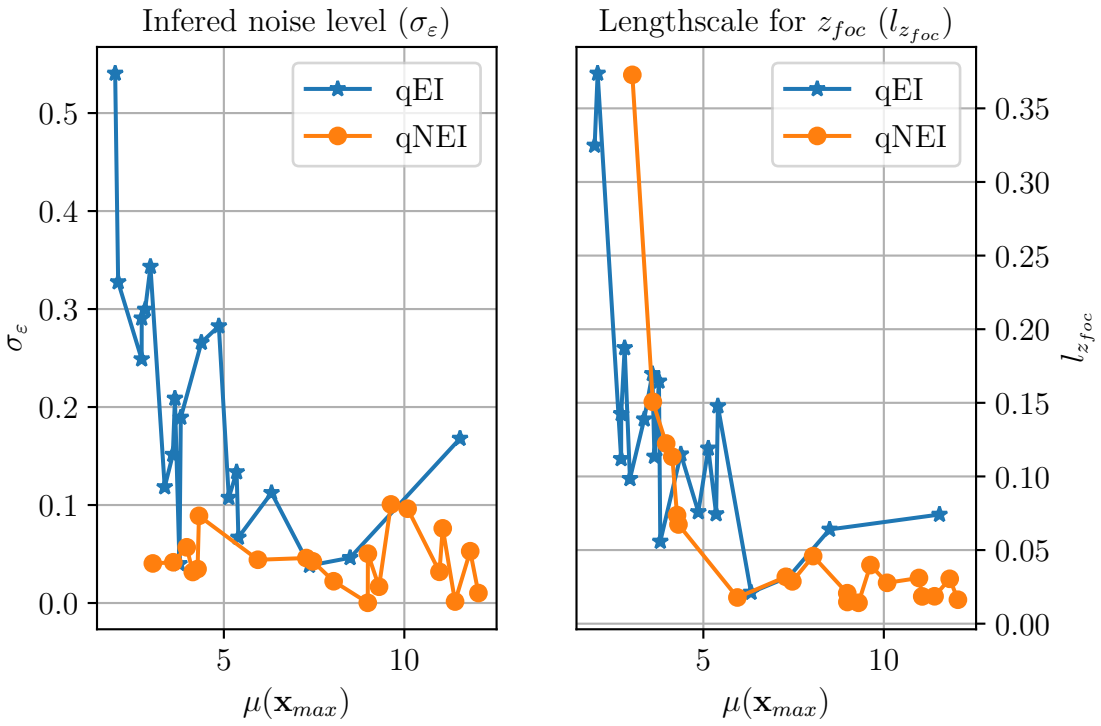

Figure 4.5: Inferred noise level σ_ε (left) and lengthscale for laser focus parameter $l_{z_{foc}}$ (right) as a function of the maximum mean function value $\mu(\mathbf{x}_{max})$ from BO on PIC simulated target function with qNEI and qEI.

4.2 Gaussian process regression modeling

In this section the results and conclusions of modeling the target surface with Gaussian process regression are presented. The section starts with the results from training a Multi-output model on the three output parameters charge, energy spread and median energy and then moves on to discussing training directly on the target function values. The data used for training and validation was PIC simulated data as described in Sec. 3.1.

4.2.1 Training of the Multi-output model

For the multi-output model the full base data set and the refined data set with additional points around the maximum were used. The use of the refined set drastically improved the mean square error (MSE) as one can see in Tab. 4.1. However, this was not the case for the single-output model, trained only on the target function f in Eq. (2.15). A possible cause of this difference is that the additional refined data is approximately homogeneously distributed in the output parameters Q, $\tilde{E}$ and ΔE in contrast to the target function where the refined data values are considerably higher than most of the data points in the base set. This causes some bias in the merged data set and thus affects the quality of the modeling. These types of modeling errors are further discussed in Sec. 4.2.2. In the multi-output model the distribution of data in the output parameters were not affected by adding the

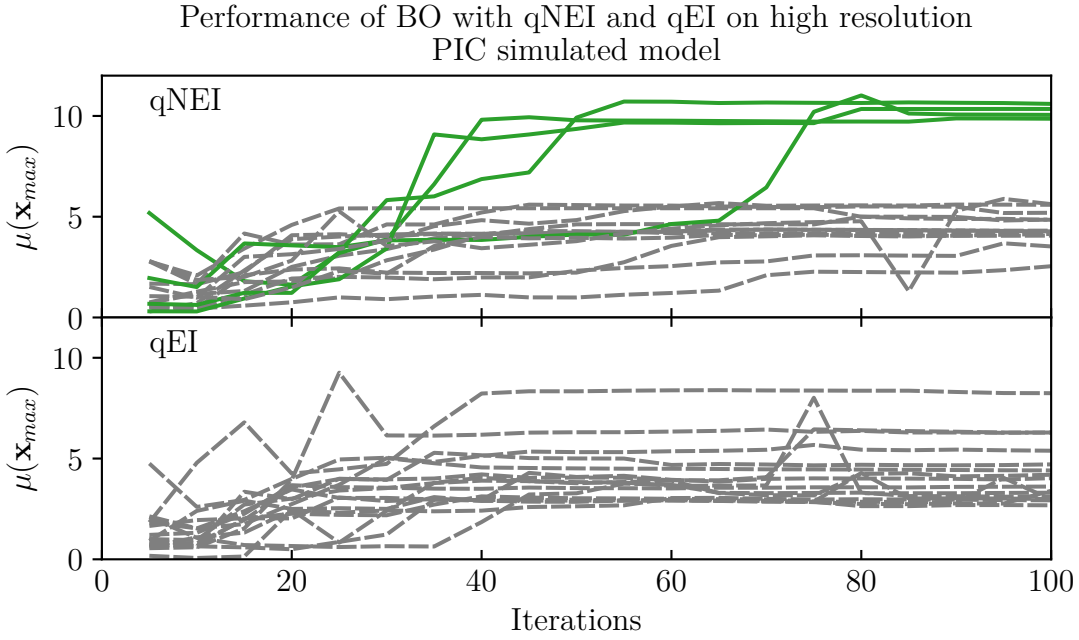

Figure 4.6: Bayesian optimization for PIC simulations using qNEI (upper graph) and qEI acquisition function. Four BO runs that found substantially higher target function values have been highlighted with green solid lines. Five initial points were used followed by 95 BO iterations.

refined points and as a consequence adding these points decreased the mean square error (MSE), therefore this set was chosen as training set. Further the multi-output model infers noise on the training data since this improved the MSE and the R^2-value. In Tab. 4.1 there is a comparison between model 1, trained with the base set and the refined set, and model 2, trained with the base set and 10 points from the refined set. Note that the MSE is considerably lower on all parameters for model 1 except for the target value. The target value was calculated from the model output, i.e. it was not modeled by a Gaussian process. In Fig. 4.7 the validation plots for model 1 (see Tab. 4.1) are shown for the parameter outputs charge, energy spread and energy median. The target function was calculated from these three outputs and used as prediction. Note that the parameter with worst accuracy is the energy spread. In general this parameter is harder to predict since it is a statistical measure

Table 4.1: MSE for model 1 trained on the refined data set and model 2 trained on the base data set and 10 points from the refined set. Both models were trained with inferred noise level.

Parameter	Model 1	Model 2
Q	349.12	1493.69
$\tilde{E}$	154.16	944.90
ΔE	42.62	153.11
Target function	0.468	0.238

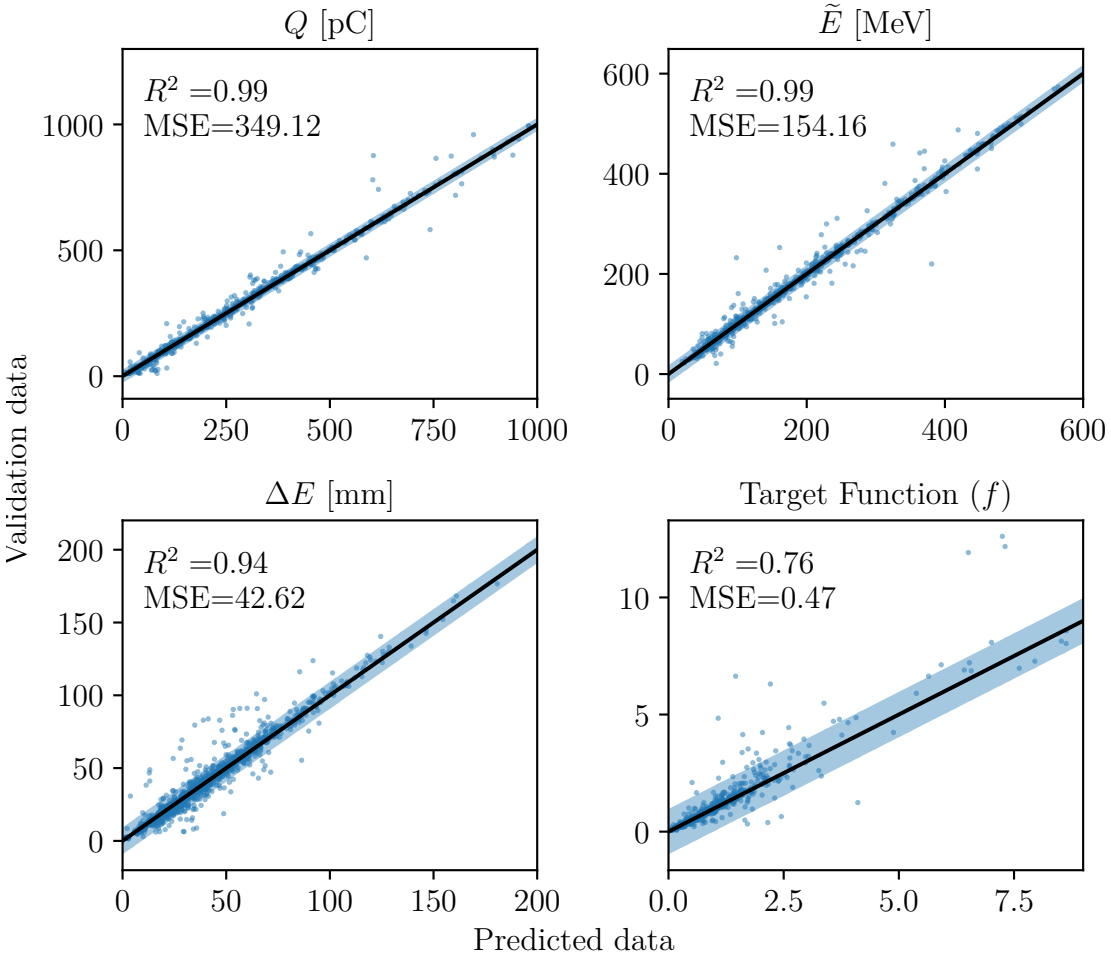

Figure 4.7: Performance of the Gaussian process model on the validation set. The reference value R^2 is the coefficient of determination. The black line is the center line $y = x$. The 94.5% confidence interval is marked by the blue shadowed area, which was approximated by calculating the standard deviation of the orthogonal distance to the center line for each point.

that varies a lot with the shape of the energy distribution of the electrons.
In Fig. 4.8 the correlation between output parameters focus position is shown. The plot was made by introducing noise as in Tab. 3.1 to the input parameters in one working point. The working point was chosen by sampling around the maximal point of the data set and then taking the coordinate of the sample yielding the highest target value.

The multi-output model creates good physical results in the sense that one can compare the distributions from Fig. 4.8 with experimental results (see [9]). Further, the discussion in Sec. 4.1 concerning the results in Fig. 4.4 supports the model. Specifically, one can see that by varying the focus position z_{foc} the energy spread deviates around one minimum (i.e. the v-shaped structure), the median energy $\widetilde{E}$ has a positive correlation and the charge Q a negative correlation. The exact same relationships between the output parameters and the focus position can be seen in the

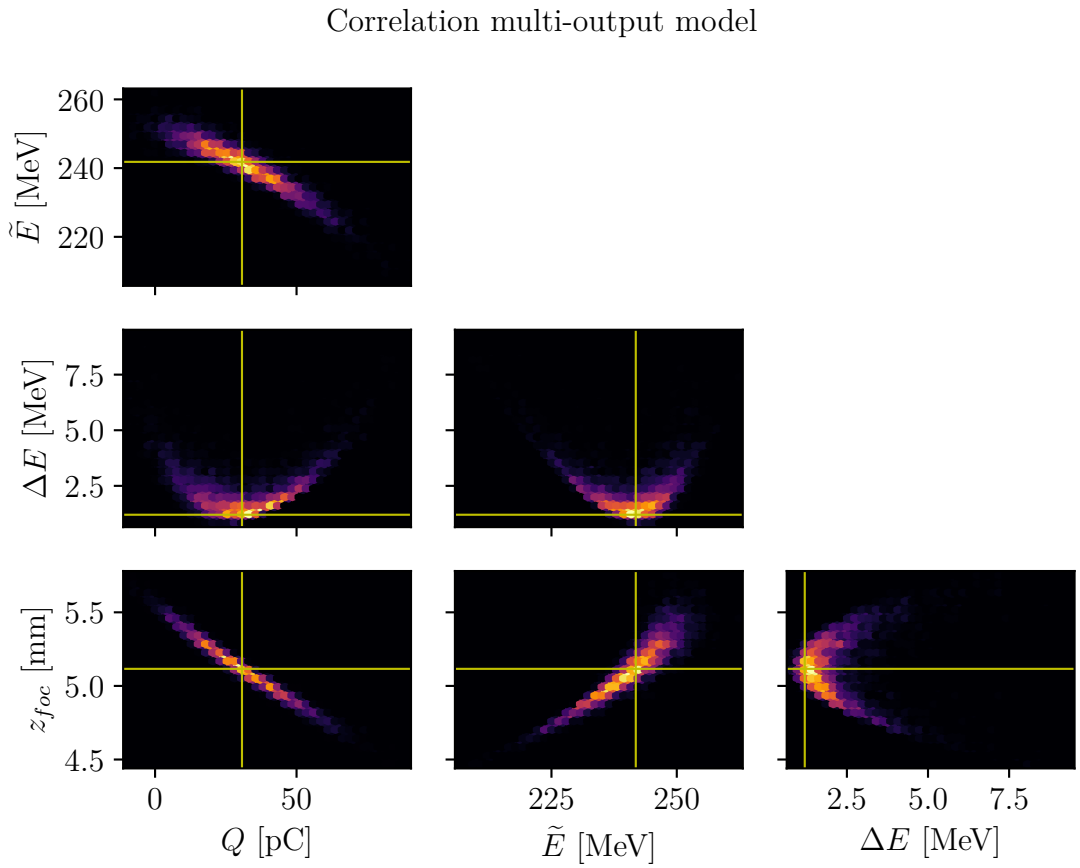

Figure 4.8: Correlation of charge, median energy, energy spread and focus point of the multi-output model around a maximal working point. The noise was modeled by a normal distributions with standard deviation as in Tab. 3.1. An absolute jitter of 0.2 in focus position was added. The yellow lines marks the working point. The optimal working point lies close to zero energy spread. This correlation plots can be compared and validated through experimental data (see [9]) and the same correlation between z_{foc} and the output parameters can also be deduced from Fig. 4.4.

PIC simulations of Fig. 4.4 with different focus positions. However, when combining the three outputs of the multi-output model the target function becomes uneven and troublesome to perform BO on. Sampling the surface one can encounter values on the order of $\sim 10^3$ in places where the energy spread is close to zero, which clearly is not a physically reasonable value. This problem is not present when training single-output models directly on the objective data since the training output data ranges from 0 to ~ 12. Thus, to avoid the points of high target values (> 100) in the multi-output model a single-output model was used for testing BO algorithms. Further the MSE of the single-output model is in the same range or better than the MSE of the objective value of the multi-output model, as can be seen by comparing Tab. 4.2 and 4.1. The building and training of this model is the subject of the next section.

4.2.2 Training of the Single-output model

Models were built using both fixed noise and inferred noise. For the inferred noise models the models were built in GPyTorch since this gave more insight about the evolution of the loss function. The fixed noise model was built with BoTorch using the *FixedNoiseModel* object. For training, the base set and refined set defined in Sec. were used. In Tab. 4.2 one can see that when adding portions of the refined set to the training data the R^2-score increases for the models with inferred noise. This is probably a consequence of the fact that the R^2-score increases with the variance of the training data and by adding data that are considerably higher than all the other data this is exactly what happens. One can also see that the MSE-value is not decreasing with more refined data indicating that the increased R^2-score has more to do with the increased variance than a better model. This trend can not be seen in the models trained with fixed noise level. One possible cause that the MSE is increasing is that the model is overfitted since it is trained with a fixed noise level low enough to be regarded as no noise, i.e. the model is more or less forced to go through every data point. This is in our case preferable since we assume that the input data is noise free but a too stiff model can also push the lengthscales of the Gaussian process into smaller values making the model overfitted. In Eq. 3.2 one can see that the R^2-score is decreasing with the MSE and if the variance of the data is increasing at the same rate as the MSE this will leave the R^2-score unchanged. One can also see that the MSE is increasing with the amount of refined data. The models with inferred noise levels might also be subject to overfitting but it is less likely since the model accepts some offset of the output data to the mean function ($\mu(\mathbf{x})$) of the model. It should also be noted that this analysis is heavily influenced by which data points the validation and training data sets include. To more confidently be able to choose the best model, cross-validation is a useful tool which has not been used here. By the reasoning presented in this section the R^2-score was disregarded and the model was chosen based on the MSE-value. Model 5 with inferred noise was chosen as the final single-output model, with the hope of avoiding some effects of overfitting.

In Fig. 4.9 the correlation of the validation set and the predictions of the model is shown. By comparing the evolution of BO algorithms on the surrogate surface with that on PIC simulations it is evident that the surrogate surface lack some details that the PIC simulated surface has. This is not strange since (1) the surrogate surface is made from samples from the PIC simulated surface and (2) there are inherent errors when modeling surfaces with Gaussian process regression. This includes defects like overfitting, underfitting and bias. Note for example that larger target values are less effectively captured by the single-output model in Fig. 4.9. This could be an effect of these values being more sparse (i.e. the output data set is biased) making the model interpret them as noise to a large extent but it could also be an effect of a more profound property of Gaussian process regression, namely the inversion of the covariance matrix. For making predictions with a Gaussian process model it is necessary to calculate or approximate the inverse of (part of) the covariance matrix. For large amounts of data this matrix is usually ill-conditioned, especially if the data points are close [23], which makes it hard to find a good approximation

Table 4.2: Training scores of single-output models. Model 1-3 were made with the BoTorch class FixedNoiseGP and model 4-6 with an ExactGP. The GPyTorch models were trained under various amounts of epochs $(100 - 1000)$ but the convergence was checked for all. For some models data around the maximum that was discovered by random sampling was added in a way to refine the model around this region. The amount of these refined data points is the second term of the sum in the data points column.

Model number	Noise-level	Data points	R^2-score	MSE
1	10^{-4}	8514+10	0.782	0.226
2	10^{-4}	8514+50	0.724	0.313
3	10^{-4}	8514+161	0.730	0.657
4	inferred	8514	0.514	0.272
5	inferred	8514+10	0.571	0.217
6	inferred	8514+161	0.830	0.335

to the inverse of the covariance matrix. There are various ways to do this, most of them involving some form of decomposition and preconditioning of the covariance matrix. For a substantial amount of data the approximations in essence become cruder resulting in larger the prediction errors. There are several techniques for mitigating this, two examples of techniques one can use for handling large amounts of data are *variational techniques* and *low rank approximations* of the covariance matrix, the GPYtorch based code used in this thesis utilizes the former. There are also local techniques that in essence split up the space and model each area with different Gaussian processes. Evaluating how these different techniques would affect the modeling falls outside the scope of this project. Nevertheless, for improving the surrogate model this would be a topic of further investigation.

There is undoubtedly more work to be done to make a model that fits the PIC data optimally. Within this project the main goal with Gaussian process regression modeling was to compare different types of BO algorithms on a target surface similar to the experimental setup. An important part was to create an algorithm for generating realistic noise on that surface. For this we deemed that the single-output model serves its purpose and we will now move on to discuss the results of the BO optimizations on this surrogate surface.

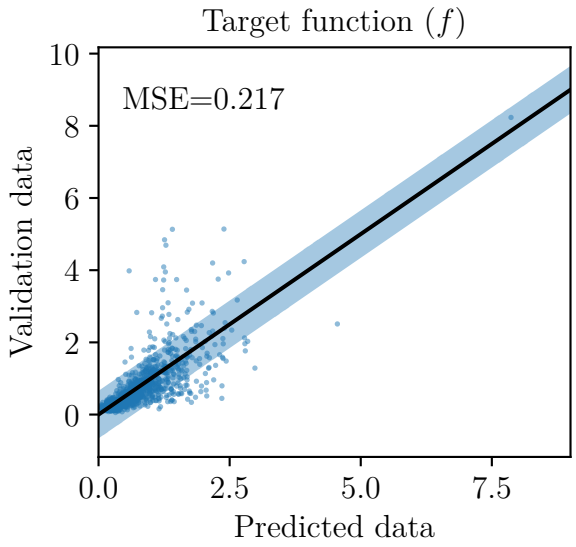

Figure 4.9: Correlation of the validation set and the predictions of the validation set made with the single-output model. The mean square error and coefficient of determination (R^2-value) has been calculated. Note that the coefficient of determination is slightly higher and the mean square error is lower than the corresponding values for the multi-output model.

4.3 Bayesian optimization on Single-output model

In this section the analysis of different BO algorithms will be presented. The comparison has been made on the surrogate surface since this is a much cheaper way to evaluate the function surface than by PIC simulations. A quantitative comparison of the two acquisition functions qNEI and qEI is made. In experiments the pulse duration and spot size of the laser is approximately 16 µm and 29 fs respectively. For the analysis here we choose to set them to the corresponding maximum target value of the training set, this was a pulse duration of 30.90 fs and spot size of 14.99 µm. In this way the surface is guaranteed to contain this maximum.

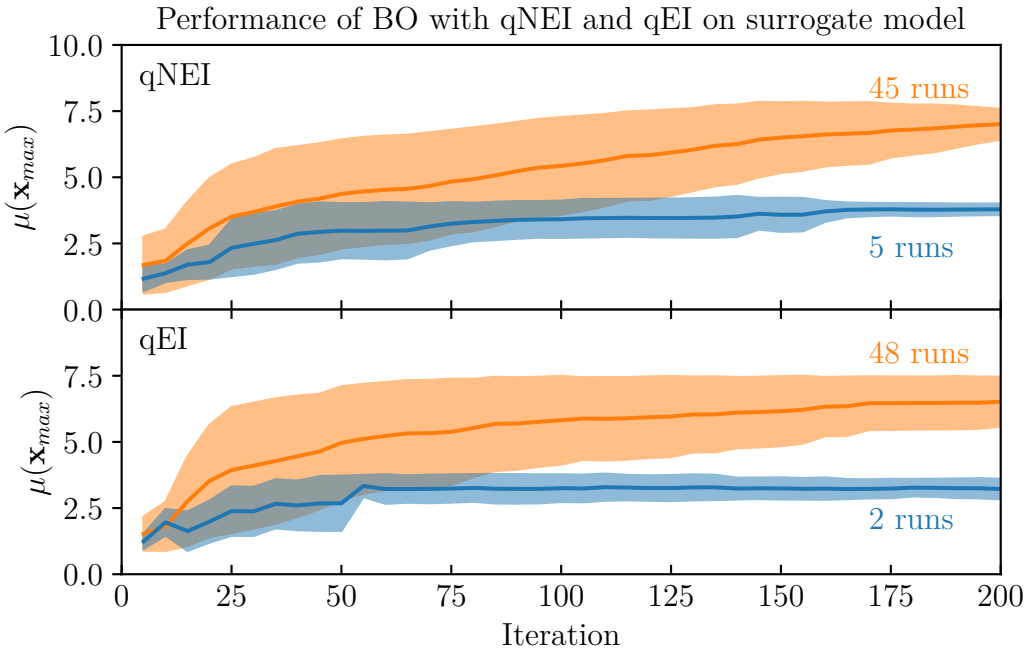

Figure 4.10: Convergence of BO with acquisition function qNEI and qEI. The plots shows the max of the models mean function $\mu(\mathbf{x_{max}})$. The runs converged to two different maxima these are marked as blue and orange and the corresponding $\mathbf{x}_{max}$ coordinates can be seen in Fig. 4.11. An output noise level of absolute value 0.4 was used corresponding to roughly 5% of the maximum value of the surface. 50 runs were made in total and the number of runs converging to the higher maximum for qNEI and qEI were 45 respective 48. The extraction of $\mu(\mathbf{x}_{max})$ and $\mathbf{x}_{max}$ were only made every fifth iteration since the calculation was relatively costly and because of this the data starts at iteration 5.

In Fig. 4.10 the evolution of BO with acquisition function qEI and qNEI is shown. One can see that two maxima are found, one maximum approximately twice as large as the other. The different maxima were calculated through the *K-means clustering algorithm* which is an algorithm that minimizes the square difference between the data points and the center of the cluster. It is clear that qNEI has a stronger convergence than qEI and in Tab. 4.3 one can see that qNEI converge to higher maximum. One can also see that qEI compared to qNEI has a larger spread of the suggested maximum, corresponding to $\mu(\mathbf{x}_{max})$ of the posterior in the last BO iteration, which shows that BO with qNEI converge towards the maximum with higher precision.

Fig. 4.11 shows the convergence of the input parameters during BO. Comparing the graphs one can possibly see that qEI converges quicker for the blue set of runs. This is supposedly a consequence of that qEI views the target function values as if they did not have any noise, which means that the algorithm will value high target function values higher than qNEI. The latter acquisition function on the other hand assigns the target function values the maximum value sampled from the posterior predictive of the surrogate model at the input parameters sampled so far. The

consequence is that the qEI values a previously sampled high value of the target function higher than qNEI in the context of it being the global maximum. This can make the function converge quicker but can also make it less accurate as in this case where BO with qNEI converges to a smaller interval of the search space than BO with qEI which can be seen in Tab. 4.3. Further, this maximum has a lower mean over the different runs, see Fig. 4.10, which indicates that fewer BO runs with qEI found a good maximum. However, there are too few runs converging to the lower maximum to draw proper conclusions about this.

Table 4.3: The mean and standard deviations of the final $\mu(\mathbf{x}_{max})$ and $\mathbf{x}_{max}$ suggested by the different BO runs with qNEI and qEI. The cumulative max, i.e. the maximal value sampled from the target function during one run is also included as a mean over all runs. 50 runs where made in total and 45 and 48 converged to this higher maximum for qNEI and qEI respectively. The maximum and its deviations found with the BO using 20 measurements at each set point (multi-shot) is also included in the table. The multi-shot BO finds a maximum comparable to the one found with the single-shot BO (qNEI in table) in just 95 iterations instead of 195 as for the single-shot algorithms.

Parameter	qNEI	qEI	Multi-shot
Cumulative max	7.95 ± 0.54	7.26 ± 0.78	7.52 ± 0.83
$\mu(\mathbf{x}_{max})$	7.03 ± 0.54	6.54 ± 0.90	7.45 ± 0.84
E_{laser} [J]	2.49 ± 0.06	2.51 ± 0.08	2.46 ± 0.08
z_{foc} [mm]	4.82 ± 0.11	4.81 ± 0.15	5.15 ± 0.08
c_{N_2} (%)	24.9 ± 0.6	25.1 ± 0.8	12.3 ± 0.8
n_0 [10^{18}cm^{-3}]	0.632 ± 0.037	0.616 ± 0.040	0.624 ± 0.122
Amount of runs	45	48	44
BO iterations	195	195	95

It is no easy task to characterize a six-dimensional surface, but from how the BO converges it can be concluded that there are two prominent maxima. Comparing the evolution of BO on this surface to the evolution on the PIC simulated surface it is clear that the latter is much more detailed. This is not surprising since the modeled surface is built by fitting a GP to sampled points of the PIC simulated data and suffers from modeling defects previously discussed. The modeled surface clearly lacks some details that the PIC-simulated surface has. Even so it suffices to try different algorithms on. The qNEI algorithm is more precise at finding the maximum in an area close to the maximum but more runs of the qNEI algorithm also found the false maximum compared to the qEI algorithm. The qNEI also has a higher final mean value of the runs than the qEI. The same result is shown by BO on the PIC simulated surface as one can see in Fig. 4.6. Supported by this results it was decided that for the following BO algorithms qNEI was used as acquisition function.

The system at LUX has a 1Hz repetition rate. Changing working point takes approximately one minute. This means that it is reasonable to gather more data than one measurement from each working point. In Fig. 4.12 the input to the BO algo-

rithm is a mean of 20 measurements at each set point. In this way some of the noise is averaged away and the BO algorithm has a faster convergence. Note that here only 100 BO iterations were made compared to 200 in Fig. 4.10 and 4.11.

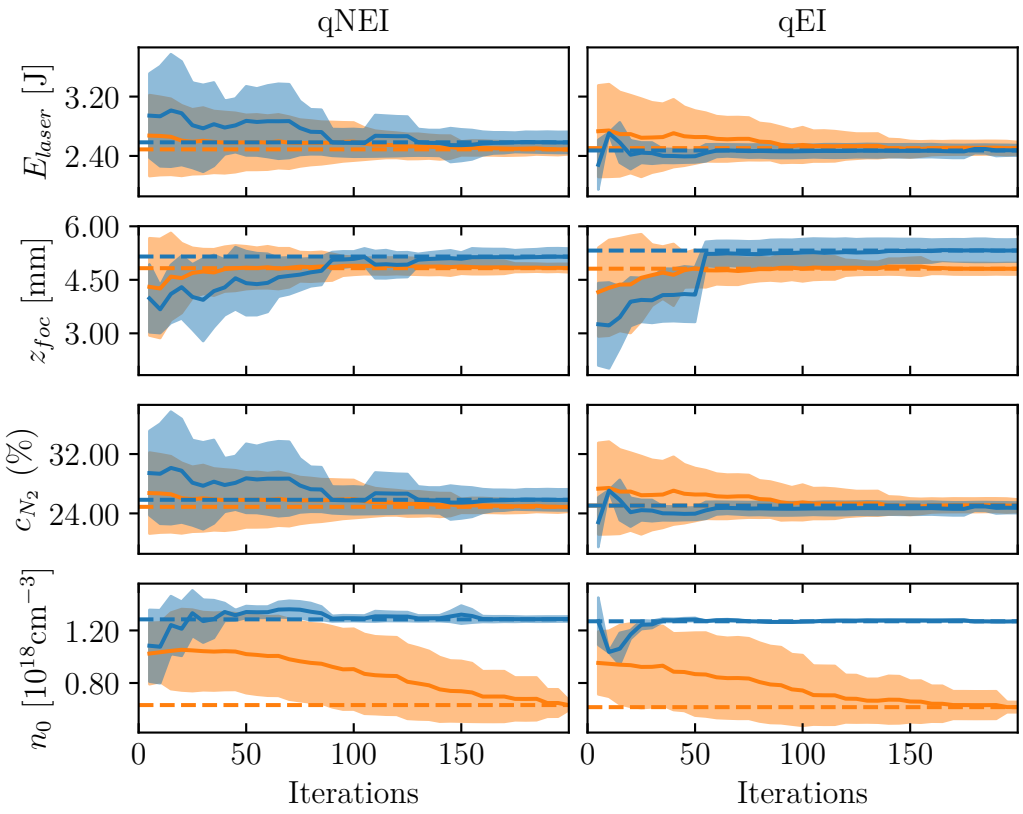

Figure 4.11: Shows the convergence the of input parameters corresponding to the suggested maximum $\mu(\mathbf{x}_max)$ of BO with acquisition function qEI and qNEI. The runs are divided into the BO runs converging to two separate maxima. The orange trajectories correspond to a higher maximum. An output noise level of absolute value 0.4 was used corresponding to roughly 5% of the maximum value of the surface. The blue data set consists of 4% of the runs for qEI and 10% of the runs for qNEI. 50 runs were made in total. The extraction of $\mu(\mathbf{x}_{max})$ and $\mathbf{x}_{max}$ were only made every 5:th iteration since the calculation was relatively costly. Because of this the data starts at iteration 5.

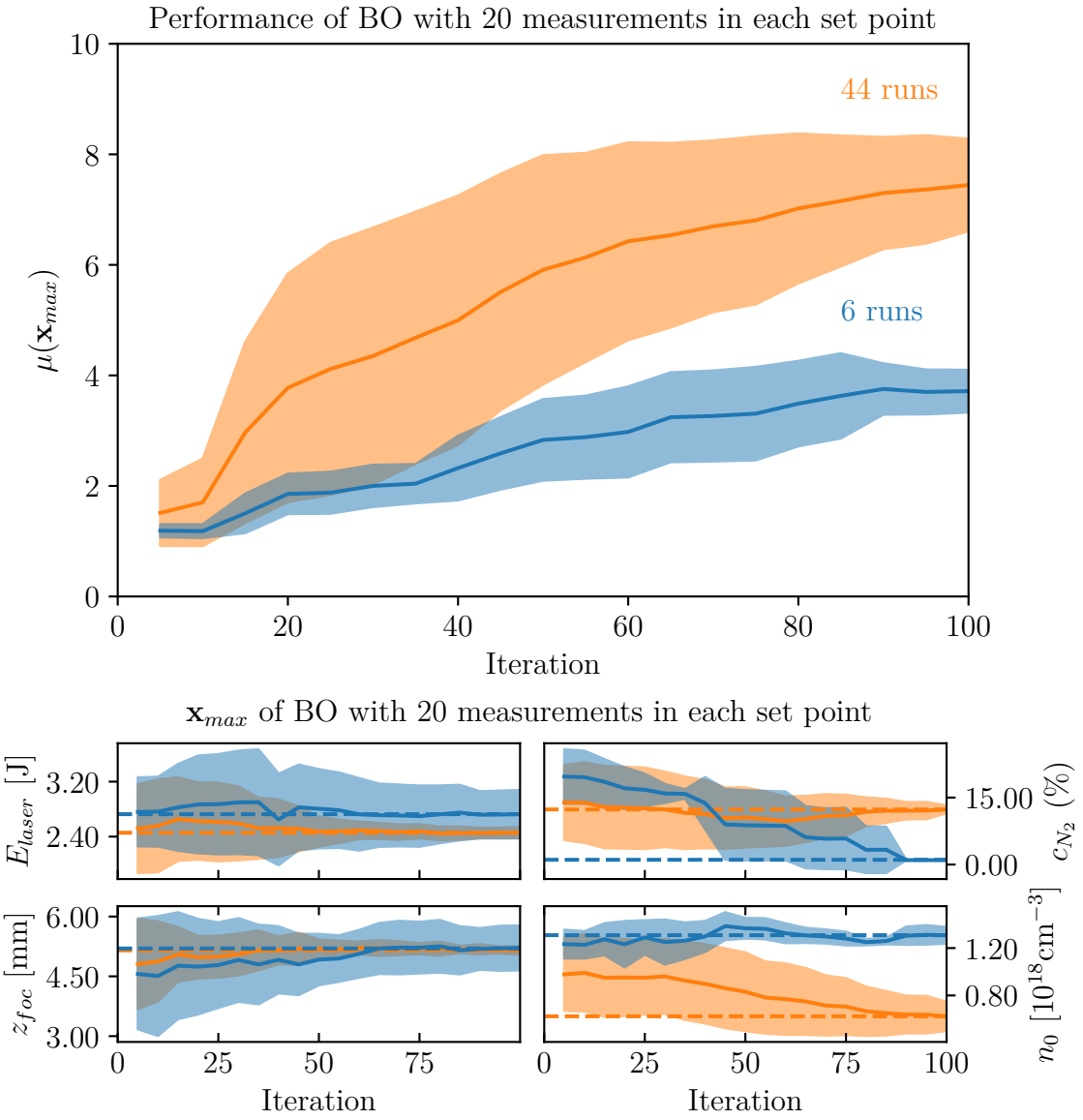

Figure 4.12: The upper graph shows the performance of BO on the surrogate model with 20 measurements in each point. The mean value of these 20 measurements both for the input and output parameters were given as input to the BO algorithm. Note that here only 95 BO iterations were enough to find the maximum that was found in the single shot case with 200 iterations. The lower graph shows the evolution of input coordinate corresponding to maximal predicted point. Note that $\mathbf{x}_{max}$ converge much faster than for the single-shot BO algorithm in Fig. 4.11.

5
Conclusion

Within this study a thorough evaluation of the use of Bayesian optimization (BO) for acquiring good beam quality in a laser plasma accelerator has been realized. In doing so, Particle-In-Cell (PIC) simulations were used for modeling of laser-plasma interaction in the accelerating process. Further, with PIC simulations as input Gaussian process regression was exploited for creation of a model that could be used for fast evaluation of the target surface.

The result shows that BO using q-Noisy Expected Improvement (qNEI) as acquisition function performed better than BO using q-Expected Improvement (qEI). However in noisy environments (i.e. noise levels of $\sim 5\%$) the convergence on the target surface were typically not satisfactory to the extent that one BO (with 100 iterations) was enough to find an adequate maximum. For BO with 100 iterations on the high resolution PIC simulated surface $\sim 1/4$ found a high maximum. By comparison, on the single-output model $\sim 90\%$ found a high maximum after 200 BO iterations. The convergence was greatly improved by taking several measurements at each set point. For a BO optimization with 20 measurements at each set point the convergence was approximately 80% of all runs within only 100 BO iterations. For the system at LUX, with a repetition rate of 1 Hz, 20 measurements at each point is reasonable. In the next generation of the LUX experiment, a repetition rate of 1000 Hz is intended and thus more measurements at each set point would be possible. This would improve the statistics used as input for the BO and could lead to better convergence.

In the context of BO there is more that can be done to better adapt to a noisy environment. The analysis in this thesis was limited to the two acquisition functions qEI and qNEI. The latter were deemed to be the most promising through theoretical research, nevertheless there are several other acquisition functions one could consider. In Gaussian process regression noise is accounted for through the basic assumptions of the model and optimized through the expression of the marginal likelihood. The noise is assumed to be normally distributed because of computational advantages and because this is often a good approximation. Nevertheless, there are also models built on variational principles that handle non-Gaussian noise distributions. Further, most commonly the noise is assumed to be *homoscedastic*, meaning that the standard deviation of the noise is equal every where in space. It is possible to account for the opposite case *heterorscedastic* noise, this has however not been evaluated in this study.

For a large extent of the evaluations of BO a so called surrogate model was used. The building and training of this model was extensively evaluated. Two different types of models were built. One multi-output model for which the functional surface showed characteristics that could be explained by the dynamics of plasma interaction. This helped to establish the validity of the model. In addition the distribution of data generated by this model could be compared with that of previous experimental data and showed good resemblance. Despite this the multi-output model could not be used for evaluation of BO algorithms foremost because of uncertainties in the energy spread which made the target value fluctuate between unphysical values. Therefore a single output model was used instead. Nevertheless the study shows that a multi-output or single-output model could be used for other purposes when (fast) modeling of the characteristics of the electron beam is needed. Possible fruitful topics of research are finding use cases for such a model. Both the single- and multi-output model would however benefit from detailed research on the implementation of Gaussian process regression with an extensive amount of data.

In summary, the study shows that BO with acquisition function qNEI is to prefer over qEI for optimization of beam quality in a laser-plasma accelerator. Further, the study shows that modeling the laser plasma interaction with Gaussian process regression is feasible and yields a, to some extent, physically valid model. Finally, it shows that by taking 20 measurements at each point, faster and more precise convergence can be reached.

Bibliography

[1] Z. Huang, Y. Ding, and C. B. Schroeder *Compact X-ray Free-Electron Laser from a Laser-Plasma Accelerator Using a Transverse-Gradient Undulator* Phys. Rev. Lett. 109, 204801 (2012)

[2] M. Fuchs, R. Weingartner, A. Popp et al. *Laser-driven soft-X-ray undulator source* Nature Phys 5, 826–829 (2009)

[3] A. R. Maier, A. Meseck, S. Reiche, C. B. Schroeder, T. Seggebrock, and F. Grüner *Demonstration Scheme for a Laser-Plasma-Driven Free-Electron Laser* Phys. Rev. X 2, 031019 (2012)

[4] Cameron G. R. Geddes et al. *Compact quasi-monoenergetic photon sources from laser-plasma accelerators for nuclear detection and characterization* Nucl. Instrum. Methods Phys. Res., Sect. B 350, 116 (2015)

[5] A. R. Maier et al. *Decoding Sources of Energy Variability in a Laser-Plasma Accelerator*. Phys. Rev. X 10, 031039 (2020)

[6] A. R. Maier, M. Kirchen, F. Grüner. *Brilliant Light Sources driven by Laser-Plasma Accelerators* Springer International Publishing, (2016)

[7] E. Esarey, C. B. Schroeder, and W. P. Leemans *Physics of laser-driven plasma-based electron accelerators* Rev. Mod. Phys. 81, 1229 (2009) Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA

[8] A. J. Gonsalves et al. *Petawatt Laser Guiding and Electron Beam Acceleration to 8 GeV in a Laser-Heated Capillary Discharge Waveguide* Phys. Rev. Lett. 122, 084801 (2019)

[9] M. Kirchen et al. *Optimal Beam Loading in a Laser-Plasma Accelerator*. PhysRevLett.126.174801, (2021)

[10] R. Fitzpatrick *Plasma Physics* CRC Press 2014, Boca Raton

[11] E. Esarey, C. B. Schroeder, and W. P. Leemans *Physics of laser-driven plasma-based electron accelerators* Rev. Mod. Phys. 81, 1229, (Published 27 August 2009) Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA

[12] Sprangle, P. and Esarey, E. and Ting, A. *Nonlinear theory of intense laser-plasma interactions* PhysRevLett.64.2011, 1990

[13] A. G. R. Thomas *Acceleration of Electrons in Plasma* Proceedings of the 2019 CERN–Accelerator–School course on High Gradient Wakefield Accelerators, Sesimbra, (Portugal)

[14] V I Berezhiani and I G Murusidze *Interaction of highly relativistic short laser pulses with plasmas and nonlinear wake-field generation* Physica Scripta, vol.45, num.2, (1992)

[15] Peter I. Frazier. *A Tutorial on Bayesian Optimization*. arXiv:1807.02811 [stat.ML] (2018) p.1-3.

[16] C. E. Rasmussen, C. K. I. Williams *Gaussian Processes for Machine Learning* MIT Press 2006, Massachusetts Institute of Technology

[17] Jian Wu. *KNOWLEDGE GRADIENT METHODS FOR BAYESIAN OPTIMIZATION.* https://people.orie.cornell.edu/jdai/thesis/JianWuThesis.pdf. (2017) Cornell University (Ithaca). (p.10-12)

[18] The code is largely based on the paper *BOTORCH: A Framework for Efficient Monte-Carlo Bayesian Optimization* by M. Balandat et al. *Acquisition Function APIs.* https://botorch.org/api/acquisition.html. Facebook Open Source (2021)

[19] B. Letham et.al *Constrained Bayesian Optimization with Noisy Experiments* Bayesian Analysis, vol. 14 (2017)

[20] M. Balandat et al. *BOTORCH: A Framework for Efficient Monte-Carlo Bayesian Optimization.* arXiv:1910.06403 [cs.LG]. Advances in Neural Information Processing Systems 33 (2020)

[21] Rémi Lehe et al. *A spectral, quasi-cylindrical and dispersion-free Particle-In-Cell algorithm.* https://www.sciencedirect.com/science/article/pii/S0010465516300224 Computer Physics Communications, vol. 203, p.66-82 (2016)

[22] A. Pukhov. *Particle-In-Cell Codes for Plasma-based Particle Acceleration.* https://cds.cern.ch Published by CERN in the Proceedings of the CAS-CERN Accelerator School: Plasma Wake Acceleration, Geneva, Switzerland, 23–29 November 2014, edited by B. Holzer, CERN-2016-001 (CERN, Geneva, 2016)

[23] H. Mohammadi et.al *An Analytic Comparison of Regularization Methods for Gaussian Processes* arXiv:1602.00853 [math.OC] (2017)

A

Appendix 1

A.1 Derivation of ponderomotive force and nonlinear poisson equation

The fluid momentum equation 2.5 can be expressed using the electrostatic potential Φ and magnetic vector potential $\mathbf{A}$ as

$$\frac{\partial \mathbf{p}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{p} = e\left(\frac{\partial \mathbf{A}}{\partial t} + \nabla\Phi - (\mathbf{v} \times \nabla \times \mathbf{A})/c\right). \tag{A.1}$$

Using the identity

$$\begin{aligned} \nabla \mathbf{p}^2 &= 2\left((\mathbf{p} \cdot \nabla)\mathbf{p} + \mathbf{p} \times (\nabla \times \mathbf{p})\right) \\ \Rightarrow \mathbf{p} \cdot \nabla \mathbf{p} &= \frac{1}{2}\nabla \mathbf{p}^2 - \mathbf{p} \times (\nabla \times \mathbf{p}) \\ \Rightarrow \mathbf{v} \cdot \nabla \mathbf{p} &= \frac{1}{2m\gamma}\nabla \mathbf{p}^2 - \mathbf{v} \times (\nabla \times \mathbf{p}) \end{aligned} \tag{A.2}$$

where $\mathbf{p} = m\gamma\mathbf{v}$ have been used, in Eq. (A.1) yields

$$\frac{\partial \mathbf{p}}{\partial t} = e\nabla\Phi + e\frac{\partial \mathbf{A}}{\partial t} - \frac{1}{2m\gamma}\nabla \mathbf{p}^2 + \mathbf{v} \times \nabla \times (\mathbf{p} - (e/c)\mathbf{A}) \tag{A.3}$$

1D approximation

In a 1D approximation the plasma quantities and the electromagnetic potentials are functions of z and t only, i.e. $\Phi = \Phi(z,t)$, $\mathbf{A} = \mathbf{A}(z,t)$, $\mathbf{p} = \mathbf{p}(z,t)$, and the derivatives with respect to x and y is thus zero. The last term on the right hand side vanishes due to conservation of transverse canonical momentum of the electrons which, if the initial transverse canonical momentum is assumed to be zero (before the laser enters the plasma), implies that $\mathbf{p}_\perp = (e/c)\mathbf{A}_\perp$. This can be seen from

$$\nabla \times (\mathbf{p} - (e/c)\mathbf{A}) = \nabla \times ((\mathbf{p}_\perp + \mathbf{p}_{||}) - (e/c)(\mathbf{A}_\perp + \mathbf{A}_{||})) = \nabla \times (\mathbf{p}_{||} - (e/c)\mathbf{A}_{||}) \tag{A.4}$$

where the last equality is zero in the 1D approximation due to vanishing transverse derivatives (the nabla operator and $\mathbf{p}_{||} - (e/c)\mathbf{A}_{||}$ are then parallel). This yields

$$\frac{\partial \mathbf{p}}{\partial t} = e\nabla_{||}\Phi + e\frac{\partial \mathbf{A}}{\partial t} - \frac{1}{2m\gamma}\nabla_{||}\mathbf{p}^2 \tag{A.5}$$

where $\nabla_{||} = (0, 0, \partial_z)$. The longitudinal component of the 1D approximation of the fluid momentum equation in Eq. (A.5) is

$$\begin{aligned} \frac{\partial \mathbf{p}_{||}}{\partial t} &= e\frac{\partial \Phi}{\partial z} + e\frac{\partial \mathbf{A}_{||}}{\partial t} - \frac{1}{2m\gamma}\frac{\partial \mathbf{p}^2}{\partial z} \\ &= e\frac{\partial \Phi}{\partial z} + e\frac{\partial \mathbf{A}_{||}}{\partial t} - \frac{1}{2m\gamma}\left(\frac{\partial \mathbf{p}_{\perp}^2}{\partial z} + \frac{\partial \mathbf{p}_{||}^2}{\partial z}\right) \\ \Rightarrow \frac{\partial \mathbf{p}_{||}}{\partial t} + \frac{1}{2m\gamma}\frac{\partial \mathbf{p}_{||}^2}{\partial z} &= e\frac{\partial \Phi}{\partial z} + e\frac{\partial \mathbf{A}_{||}}{\partial t} - \frac{1}{2m\gamma}\frac{\partial \mathbf{p}_{\perp}^2}{\partial z}. \end{aligned} \tag{A.6}$$

The left hand side can be rewritten as a total time derivative

$$\frac{d\mathbf{p}_{||}}{dt} = \frac{\partial \mathbf{p}_{||}}{\partial t} + \mathbf{v}_{||} \cdot \frac{\partial \mathbf{p}_{||}}{\partial z} \tag{A.7}$$

when noticing that

$$\frac{\partial \mathbf{p}_{||}^2}{\partial z} = 2\mathbf{p}_{||} \cdot \frac{\partial \mathbf{p}_{||}}{\partial z} = 2m\gamma \mathbf{v}_{||} \cdot \frac{\partial \mathbf{p}_{||}}{\partial z} \tag{A.8}$$

which yields

$$\frac{d\mathbf{p}_{||}}{dt} = e\frac{\partial \Phi}{\partial z} + e\frac{\partial \mathbf{A}_{||}}{\partial t} - \frac{1}{2m\gamma}\frac{\partial \mathbf{p}_{\perp}^2}{\partial z} \tag{A.9}$$

For laser polarized in transverse direction, $\mathbf{A}_{||} = 0$, and we have

$$\frac{d\mathbf{p}_{||}}{dt} = e\frac{\partial \Phi}{\partial z} - \frac{1}{2m\gamma}\frac{\partial \mathbf{p}_{\perp}^2}{\partial z} \tag{A.10}$$

where $\mathbf{p}_{\perp}^2 = (e/c)\mathbf{A}^2$ due to concervation of transverse canonical momentum.

Quasi static approximation

The longitudinal component of the 1D approximation in Eq. (A.5) with $\mathbf{A} = \mathbf{A}_{\perp}$ is

$$\frac{\partial \mathbf{p}_{||}}{\partial t} = e\frac{\partial \Phi}{\partial z} - \frac{1}{2m\gamma}\frac{\partial \mathbf{p}^2}{\partial z}. \tag{A.11}$$

Making a coordinate transformation to the frame moving with the velocity of the wake, that is $\xi = z - v_p t$ and $\tau = t$, where we have partial derivatives according to

$$\frac{\partial}{\partial z} = \frac{\partial}{\partial \xi} \quad \text{and} \quad \frac{\partial}{\partial t} = \frac{\partial}{\partial \tau} - v_p\frac{\partial}{\partial \xi} \tag{A.12}$$

yielding

$$\frac{\partial \mathbf{p}_{||}}{\partial \tau} - v_p\frac{\partial \mathbf{p}_{||}}{\partial \xi} = e\frac{\partial \Phi}{\partial \xi} - \frac{1}{2m\gamma}\frac{\partial \mathbf{p}^2}{\partial \xi}. \tag{A.13}$$

In this frame, a quasistatic approximation can be made [12]. This means that we can consider the quantities constant over time τ resulting in

$$-v_p\frac{\partial \mathbf{p}_{||}}{\partial \xi} = e\frac{\partial \Phi}{\partial \xi} - \frac{1}{2m\gamma}\frac{\partial \mathbf{p}^2}{\partial \xi} \tag{A.14}$$

Noticing that the derivative of γ with respect to ξ is

$$\frac{\partial\gamma}{\partial\xi} = \frac{\partial}{\partial\xi}\left(1+\frac{\mathbf{p}^2}{(mc)^2}\right)^{1/2} = \frac{1}{2(mc)^2}\frac{\partial\mathbf{p}^2}{\partial\xi}\cdot\left(1+\frac{\mathbf{p}^2}{(mc)^2}\right)^{-1/2} = \frac{1}{2(mc)^2\gamma}\frac{\partial\mathbf{p}^2}{\partial\xi} \quad \text{(A.15)}$$

we get

$$\begin{aligned} -v_p\frac{\partial\mathbf{p}_{||}}{\partial\xi} &= e\frac{\partial\Phi}{\partial\xi} - mc^2\frac{\partial\gamma}{\partial\xi} \\ \Rightarrow \quad \frac{\partial}{\partial\xi}(v_p\mathbf{p}_{||} &+ e\Phi - mc^2\gamma) = 0. \end{aligned} \quad \text{(A.16)}$$

Integrating the left hand side yields

$$\begin{aligned} &\int_\xi^{\xi_1}\frac{\partial}{\partial\xi'}(v_p\mathbf{p}_{||}+e\Phi-mc^2\gamma)d\xi' = \\ &= v_p\mathbf{p}_{||}(\xi_1)+e\Phi(\xi_1)-mc^2\gamma(\xi_1)-(v_p\mathbf{p}_{||}(\xi)+e\Phi(\xi)-mc^2\gamma(\xi)) \\ &= -mc^2-(v_p\mathbf{p}_{||}(\xi)+e\Phi(\xi)-mc^2\gamma(\xi)) \end{aligned} \quad \text{(A.17)}$$

where ξ_1 is taken to be in front of the laser pulse where the plasma is assumed to be homogeneous and at rest and no static potential is induced by charge separation. Note that the fluid momentum is zero here[1] and thus $\gamma = 1$. We thus have

$$\begin{aligned} v_p\mathbf{p}_{||} + e\Phi - mc^2\gamma &= -mc^2 \\ \Rightarrow -e\Phi + mc^2\gamma - v_pm\gamma\mathbf{v}_{||} &= mc^2 \\ \Rightarrow -\frac{e\Phi}{mc^2} + \gamma - \frac{v_p\gamma\mathbf{v}_{||}}{c^2} &= 1 \end{aligned} \quad \text{(A.18)}$$

where $\mathbf{p}_{||} = m\gamma\mathbf{v}_{||}$ have been used. Introducing the normalized longitudinal fluid velocity $\beta = \mathbf{v}_{||}/c$ and the normalized scalar potential $\phi = \frac{e\Phi}{mc^2}$ we get

$$\begin{aligned} -\phi + \gamma - \frac{v_p}{c}\gamma\beta &= 1 \\ \Rightarrow \left(1-\frac{v_p}{c}\beta\right)\gamma &= 1+\phi. \end{aligned} \quad \text{(A.19)}$$

For $v_p = c$ this yields

$$\frac{\gamma}{1+\phi} = \frac{1}{1-\beta}. \quad \text{(A.20)}$$

The gamma factor can be written as

$$\gamma^2 = \frac{1+a^2}{1-\beta} \quad \text{(A.21)}$$

where $\mathbf{a} = e\mathbf{A}/mc^2$ is the normalized vector potential. Squaring Eq. A.20 and inserting Eq. A.21 yields

$$\begin{aligned} \frac{\gamma^2}{(1+\phi)^2} &= \frac{1}{(1-\beta)^2} \\ \Rightarrow \frac{1+a^2}{(1-\beta)(1+\phi)^2} &= \frac{1}{(1-\beta)^2} \\ \Rightarrow \frac{1+a^2}{(1+\phi)^2} &= \frac{1}{1-\beta}. \end{aligned} \quad \text{(A.22)}$$

[1] The fluid momentum is proportional to the mean velocity for the electrons at $z = \xi$ in phase space

The 1D continuity equation (2.4) in the quasistatic approximation is

$$\frac{\partial n\mathbf{v}_{||}}{\partial \xi} - v_p \frac{\partial n}{\partial \xi} = 0 \tag{A.23}$$

and with the normalized longitudinal fluid velocity it become

$$\frac{\partial}{\partial \xi} nc\left(\beta - \frac{v_p}{c}\right) = 0. \tag{A.24}$$

This is integrated in the same way as the momentum equation and with $n(\xi_1) = n_0$ and $\beta(\xi_i) = 0$ this yields

$$nc\left(\beta - \frac{v_p}{c}\right) = -n_0 v_p. \tag{A.25}$$

For $v_p = c$ we get

$$\frac{n}{n_0} = \frac{1}{1-\beta}. \tag{A.26}$$

Comparing Eq. (A.22) and Eq. (A.26) we see that

$$\frac{n}{n_0} = \frac{1+a^2}{(1+\phi)^2} \tag{A.27}$$

The 1D Poisson equation, from Gauss equation in Eq. (2.2), written with the normalized scalar potential is

$$\frac{\partial^2 \phi}{\partial \xi^2} = \frac{e^2 n_0}{mc^2 \epsilon_0}\left(\frac{n}{n_0} - 1\right) \tag{A.28}$$

inseriting eq.(A.27) yields

$$\frac{\partial^2 \phi}{\partial \xi^2} = \frac{e^2 n_0}{mc^2 \epsilon_0}\left(\frac{1+a^2}{(1+\phi)^2} - 1\right) \tag{A.29}$$

where $e^2 n_0 / mc^2 \epsilon_0 = k_p^2$.

A.2 Relativistic ponderomotive force

Writing the equation of motion in terms of the vector potential $\mathbf{A}$ (with $\mathbf{E} = -\frac{\partial \mathbf{A}}{\partial t}$)

$$\frac{d\mathbf{p}}{dt} = q\left[-\frac{\partial \mathbf{A}}{\partial t} + \frac{\mathbf{p}}{\gamma m} \times (\nabla \times \mathbf{A})\right] \tag{A.30}$$

By splitting the motion and fields into one fast oscillating part and one slow drift

$$\mathbf{p} = \mathbf{p}_D + \mathbf{p}_{osc}, \tag{A.31}$$

$$\tag{A.32}$$

where

$$\frac{1}{T}\int_T \mathbf{p}_D + \mathbf{p}_{osc} dt = \mathbf{p}_D \tag{A.33}$$

T is the period of the oscillation. This means which means that we assume that the drift field and momentum does not change significantly over one period of oscillation. Taking the time average over the equation of motion and assuming $\mathbf{A} \sim cos(\omega t + kz)$ in time, we obtain

$$\left\langle \frac{\partial(\mathbf{p}_D + \mathbf{p}_{osc})}{\partial t} + \mathbf{v}_D \cdot \nabla \mathbf{p}_D + \mathbf{v}_{osc} \cdot \nabla \mathbf{p}_{osc} \right\rangle =$$
$$= \left\langle q \left[-\frac{\partial \mathbf{A}}{\partial t} + \frac{\mathbf{p}_D + \mathbf{p}_{osc}}{\gamma m} \times (\nabla \times \mathbf{A}) \right] \right\rangle \rightarrow$$
$$\frac{\partial \mathbf{p}_D}{\partial t} + \mathbf{v}_D \cdot \nabla \mathbf{p}_D + \langle \mathbf{v}_{osc} \cdot \nabla \mathbf{p}_{osc} \rangle = \frac{q}{m} \left\langle \frac{\mathbf{p}_{osc}}{\gamma} \times (\nabla \times \mathbf{A}) \right\rangle$$

Subtracting the above equation from the equation of motion we obtain

$$\frac{\partial \mathbf{p}_{osc}}{\partial t} + [\mathbf{v}_{osc} \cdot \nabla \mathbf{p}_{osc}]_{osc} = -q \frac{\partial \mathbf{A}}{\partial t} + \frac{q}{m} \left[\frac{\mathbf{p}_{osc}}{\gamma} \times (\nabla \times \mathbf{A}) \right]_{osc}$$

where $[f]_{osc}$ is the oscillating part of f. Assuming that the field changes slowly in space i.e the wave vector $\mathbf{k}$ is relatively small compared to the changes in time. We can divide terms in order of $\nabla \mathbf{p}_D \sim \nabla \mathbf{p}_{osc} \sim \nabla \mathbf{A} \sim \varepsilon$. In the 0:th order of ε the two equations above becomes

$$\frac{\partial \mathbf{p}_D^{(0)}}{\partial t} = 0 \rightarrow \mathbf{p}_D^{(0)} = 0,$$
$$\frac{\partial \mathbf{p}_{osc}^{(0)}}{\partial t} = -q \frac{\partial \mathbf{A}}{\partial t} \rightarrow \mathbf{p}_{osc}^{(0)} = -q\mathbf{A}.$$

Using the latter relationship and $\frac{d\mathbf{p}_D^{(1)}}{dt} = \frac{\partial \mathbf{p}_D^{(1)}}{\partial t}$ in the 1:st order of ε we obtain

$$\frac{d\mathbf{p}_D^{(1)}}{dt} + \left\langle \frac{\mathbf{p}_{ocs}^{(0)}}{m\gamma} \cdot \nabla \mathbf{p}_{osc}^{(0)} \right\rangle = \frac{q}{m} \left\langle \frac{\mathbf{p}_{osc}^{(0)}}{\gamma} \times (\nabla \times \mathbf{A}) \right\rangle \rightarrow$$
$$\frac{d\mathbf{p}_D^{(1)}}{dt} = \frac{q^2}{m\bar{\gamma}} \langle -\mathbf{A} \cdot \nabla \mathbf{A} - \mathbf{A} \times (\nabla \times \mathbf{A}) \rangle = -\frac{q^2}{m\bar{\gamma}} \frac{\langle \nabla \mathbf{A}^{\,2} \rangle}{2}$$

A.3 Derivation of Poissons equation

$$\frac{\partial^2 \Phi}{\partial \xi^2} = \frac{k_p^2}{1 - \beta_p^2} \left(\beta_p \left(1 - \frac{(1 - \beta_p^2)(1 + a^2)}{(1 + \Phi)^2} \right)^{-1/2} - 1 \right) \tag{A.34}$$

Letting $v_p \to c$

$$\lim_{\beta_p \to 1} \frac{k_p^2}{1-\beta_p^2} \left(\beta_p \left(1 - \frac{(1-\beta_p^2)(1+a^2)}{(1+\Phi)^2} \right)^{-1/2} - 1 \right) = \tag{A.35}$$

$$= \left[(1-x)^{-1/2} 1 + \frac{1}{2} xa + \mathcal{O}(x^2) \right] = \tag{A.36}$$

$$= \lim_{\beta_p \to 1} \frac{k_p^2}{1-\beta_p^2} \left(\beta_p \left(1 + \frac{1}{2} \frac{(1-\beta_p^2)(1+a^2)}{(1+\Phi)^2} + \mathcal{O}((1-\beta_p^2)^2) \right) - 1 \right) = \tag{A.37}$$

$$= \lim_{\beta_p \to 1} k_p^2 \left(\frac{\beta_p - 1}{1-\beta_p^2} + \frac{1}{2} \frac{\beta_p (1+a^2)}{(1+\Phi)^2} + \mathcal{O}((1-\beta_p^2)) \right) = \tag{A.38}$$

$$= \lim_{\beta_p \to 1} k_p^2 \left(-\frac{1}{1+\beta_p} + \frac{1}{2} \frac{\beta_p (1+a^2)}{(1+\Phi)^2} + \mathcal{O}((1-\beta_p^2)) \right) = \tag{A.39}$$

$$= \frac{k_p^2}{2} \left(\frac{1+a^2}{(1+\Phi)^2} - 1 \right) \tag{A.40}$$

A.4 Knowledge gradient

By deviating from point (2) *the best target function value is restricted to a previously sampled point* it is possible to formulate the Bayes optimal function for choosing the maximal next point in an noisy environment [17]. This acquisition function is called Knowledge gradient and can be defined as

$$KG(\mathbf{x}_p, \mathbb{A}) = \mathbb{E}\left[\max_{\mathbf{z} \in \mathbb{A}} \mu_{n+1}(\mathbf{z}) | \mathbf{z}_{n+1} = \mathbf{x}_p \right] - \max_{\mathbf{z} \in \mathbb{A}} \mu_n(\mathbf{z}) \tag{A.41}$$

where $\mu_n(\mathbf{x})$ is the mean value function of the posterior predictive conditioned on n target function values and corresponding $\mathbf{x}$. Here we are maximizing the difference between the current maximal mean function value and the expectation value of the maximal mean function value conditioned that $\vec{x}_p$ was chosen as the $\mathbf{z}_{n+1}$ point to evaluate.

The set $\mathbb{A}$ could theoretically be chosen to be continuous but for tractability of the problem (computational reasons) the space $\mathbb{A}$ is usually discretized [17]. If one restricts $\mathbb{A}$ to previously sampled points and assume that the target function is noiseless the KG reduces to EI as

$$KG(\mathbf{x}_p, \mathbb{A}) = \mathbb{E}\left[\max_{\mathbf{z} \in \mathbb{A} \cup \mathbf{x}_p} \mu_{n+1}(\mathbf{z}) | \mathbf{z}_{n+1} = \mathbf{x}_p \right] - y_{max} =$$

$$= \mathbb{E}\left[\max[y_{max}, \mu_{n+1}(\mathbf{x}_p)] \right] - y_{max} = \mathbb{E}\left[(\mu_{n+1}(\mathbf{x}_p) - y_{max})^+ \right] =$$

$$= \mathbb{E}\left[(f(\mathbf{x}_p) - y_{max})^+ \right] .[17] \tag{A.42}$$

A.5 Analytical implementation EI

Assuming $f(\mathbf{x}_p) \sim \mathcal{N}(\mu_D(\mathbf{x}), K_D(\mathbf{x}, \mathbf{x})))$, i.e distributed as the posterior, one can analytically calculate the expected improvement by using $\int_{-\infty}^{a} \mathcal{N}(x; \mu, \sigma^2) dx = \Phi(a; \mu, \sigma^2)$

where Φ is the cumulative density function of a normal distribution. Using this we can solve the integral of the expectation as

$$\begin{aligned}
\mathbb{E}\left[(f(\mathbf{x}) - f^*_{max})^+\right] &= \int_{f^*_{max}}^{\infty} (f - f^*_{max})\mathcal{N}(f; \mu(\mathbf{x}), K(\mathbf{x},\mathbf{x}))df = \\
&= \int_{f^*_{max}}^{\infty} (f - f^*_{max})\frac{1}{2\pi K(\mathbf{x},\mathbf{x})} e^{-\frac{1}{2}\left(\frac{f-\mu(\mathbf{x})}{K(\mathbf{x},\mathbf{x})}\right)^2} df = \\
&= \int_{f^*_{max}}^{\infty} (f - f^*_{max})\frac{1}{2\pi K(\mathbf{x},\mathbf{x})} e^{-\frac{1}{2}\left(\frac{f-\mu(\mathbf{x})}{K(\mathbf{x},\mathbf{x})}\right)^2} df = [f = -\xi] = \\
&= -\int_{-\infty}^{-f^*_{max}} (-\xi - f^*_{max})\frac{1}{2\pi K(\mathbf{x},\mathbf{x})} e^{-\frac{1}{2}\left(\frac{-\xi-\mu(\mathbf{x})}{K(\mathbf{x},\mathbf{x})}\right)^2} d(-\xi) = \\
&= \int_{-\infty}^{-f^*_{max}} (-\xi - \mu(\mathbf{x}))\frac{1}{2\pi K(\mathbf{x},\mathbf{x})} e^{-\frac{1}{2}\left(\frac{-\xi-\mu(\mathbf{x})}{K(\mathbf{x},\mathbf{x})}\right)^2} + \\
&\quad + (\mu(\mathbf{x}) - f^*_{max})\frac{1}{2\pi K(\mathbf{x},\mathbf{x})} e^{-\frac{1}{2}\left(\frac{-\xi-\mu(\mathbf{x})}{K(\mathbf{x},\mathbf{x})}\right)^2} d(\xi) = \\
&= -K(\mathbf{x},\mathbf{x})\mathcal{N}(f^*_{max}; \mu(\mathbf{x}), K(\mathbf{x},\mathbf{x})) + (\mu(\mathbf{x}) - f^*_{max})\Phi(f^*_{max}; \mu(\mathbf{x}), K(\mathbf{x},\mathbf{x})).
\end{aligned}$$

This expression can be differentiated and used to calculate the maximum of the acquisition function.

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY