

Inverse Rendering of Fusion Plasmas: Inferring Plasma Composition from Imaging Systems

E Öztürk^{1‡}, R Akers², S Pamela^{2§}, The MAST Team², P Peers^{3||} and A Ghosh^{1¶}

¹ Department of Computing, Imperial College, Prince Consort Road, London SW7 2BZ, UK

² Culham Centre for Fusion Energy, Culham Science Centre, Abingdon OX14 3EB

³ Computer Science Department, College of William & Mary, Williamsburg, VA, 23187, USA

Abstract. In this work, we develop a differentiable rendering pipeline for visualising plasma emission within tokamaks, and estimating the gradients of the emission and estimating other physical quantities. Unlike prior work, we are able to leverage arbitrary representations of plasma quantities and easily incorporate them into a non-linear optimisation framework. The efficiency of our method enables not only estimation of a physically plausible image of plasma, but also recovery of the neutral Deuterium distribution from imaging and midplane measurements alone. We demonstrate our method with three different levels of complexity showing first that a poloidal neutrals density distribution can be recovered from imaging alone, second that the distributions of neutral Deuterium, electron density and electron temperature can be recovered jointly, and finally, that this can be done in the presence of realistic imaging systems that incorporate sensor cropping and quantisation.

Submitted to: *Nucl. Fusion*

‡ <https://www.imperial.ac.uk/people/ekin.ozturk17>

§ https://stanstash.com/spamela_profile/

|| <https://www.cs.wm.edu/~ppeers/>

¶ <https://www.imperial.ac.uk/people/abhiijeet.ghosh>

1. Introduction

In fusion research, high-speed imaging of tokamak plasma can provide useful insight about the dynamics and properties of the experiment. Accurate distributions of the neutrals, the electrons, the ions and the electron temperature allows visualisations of the plasma that matches camera images through the forward rendering of the plasma emission. Furthermore, inverting the rendering enables recovering information about the distributions from images themselves. The process of recovering a 3d representation underlying the image seen by a camera is an active research topic called inverse rendering [17] in the field of computer graphics. While methods exist for reconstructing representations such as point clouds [14], implicit surfaces [19], and neural representations [10]; for plasmas the ideal representations would be the physical quantities themselves, neutrals density, electron density and temperature, which motivates the use of inverse rendering.

Inverse rendering is a highly non-linear problem. To exploit powerful non-linear optimization algorithms, the derivatives of the objective function are needed. An active area of research in computer graphics is to leverage automatic differentiation (autodiff) to compute the gradients of scene properties through backpropagation with respect to the light transport simulation. This process is also called differentiable rendering. In this paper, we demonstrate that inverse rendering of plasma properties through differentiable rendering offers an alternative method for obtaining estimates of the underlying physical quantities such as the neutrals density, electron density and electron temperature.

To estimate the physical quantities that influence the light emission and transport, we present a differentiable rendering pipeline where we leverage Null-Scattering [12] to simulate light transport in a tokamak, and Path-Replay Backpropagation [20] for computing the gradients. Our differentiable plasma rendering pipeline enables not only physically accurate simulation of the imaging system, but also efficient optimisation of the parameters underlying the forward simulation via non-linear optimisation. As we require a renderer that is capable of automatic differentiation of the rendering algorithm itself and performance suitable for fast non-linear optimisation, we build our rendering pipeline on top of the Mitsuba 3 [9] renderer. Unlike previous rendering pipelines, such as CHERAB [5], our pipeline is capable of rendering a full-resolution (1024×1024) image on a GPU in seconds to minutes, and compute its gradients to machine accuracy in a similar amount of time with minimal memory cost.

We demonstrate that, through our rendering framework, filtered and calibrated high-speed imaging systems in the MAST and MAST-U tokamaks could be used to estimate the full poloidal distributions of electron density, electron temperature and neutrals density. While we demonstrate our method on a simplified representation of the physical distributions, a fully differentiable simulation of the plasma dynamics can exploit the gradients in our pipeline to optimize poloidal and toroidal magnetic fields, neutrals density injection and any other parameters that control the simulation itself. Furthermore, given that there currently exists no diagnostic that can provide

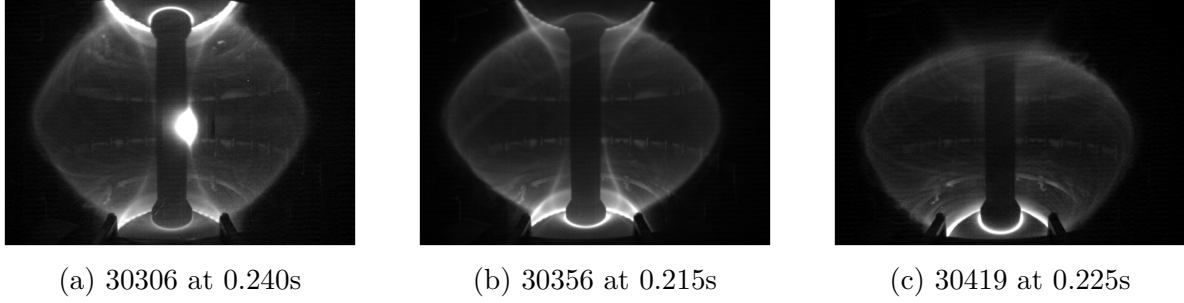


Figure 1: Images from shots 30306, 30356 and 30419 in the MAST tokamak, obtained by the Photron APX-RS (RBB) camera mounted on port HM10. These images show the variety of toroidally symmetric plasma emission configurations.

full poloidal measurements of electron density and electron temperature, nor spatial measurements of neutral density of any kind, we believe the case-study demonstrated in this paper can pave the way for a new type of diagnostic.

Previous works on reconstructing plasma properties from imaging relied on automated segmentation or hand-labelling and then inferring the properties based on known physical quantities such as the magnetic field in the vessel [7]. The closest to our work is Carr et al. [4] where the emission of a toroidally symmetric plasma is reconstructed from bolometric measurements in the MAST vessel using the Simultaneous Algebraic Reconstruction Technique (SART) and toroidal Green’s functions as the emission bases. In the graphics literature, the closest to our method belongs to Nimier-David et al. [13], where synthetic objects are reconstructed by first optimising an emission grid and then using that to infer the scattering albedo of the volume. In this work we demonstrate that similar techniques used in Nimier-David et al. [13] can be applied to fusion plasma physics using an appropriate emission model based on Photon Emissivity Coefficients (PEC) [18].

Using a poloidal map representation of the neutrals density, electron density and electron temperature as functions of flux surfaces, our optimisation method converges to a plasma composition that closely reproduces a target camera image, and 1D midplane profiles of the electron and electron temperature. We demonstrate the above principle, on simulation data from a SOLPS run on MAST [8].

In summary, our contributions are:

- Differentiable Rendering of fusion plasma,
- Framework for optimisation-based diagnostic of global plasma composition,
- A low-parameter plasma representation following the flux surfaces.

2. Methodology

The goal of this study is to demonstrate the utility of differentiable simulations for the inference of physical properties through a case-study on the reconstruction of the

distributions of the neutrals density (n_n), electron density (n_e) and electron temperature (T_e) within the MAST vessel using camera imaging and sparse measurements. Achieving this reconstruction requires physically accurate simulation of the camera imaging diagnostic in MAST, estimation of the gradients of this image with respect to the parameters of the plasma and emission model, and an algorithm that can optimise these parameters to closely reconstruct the reference image.

In the following sections, we go through the algorithmic requirements and implementation of the simulation of light emission and transport, the gradient estimation of this simulation, and define the objective function and the optimisation problem for the reconstruction of the reference image.

2.1. Overall Pipeline

In Figure 2 we show the backbone of our optimisation pipeline that we leverage for all of our experiments that includes the main objective terms. Our pipeline consists of four components:

0. the **Inputs** which consists of any parameters that are defined a-priori and not optimised, this includes the target image as well as the target measurements which are used to compute the optimisation objective,
1. the **Distributions to Optimise** which consist of the the electron density and temperature and/or neutrals density cross-sections for minimising the objective function, in Experiment i this is simply the neutrals density while in Experiments ii and iii, this is all three of the density, ionisations fraction and electron temperature,
2. the **Forward Pass**, which computes, given the distributions, the rendering of the interior of the tokamak given the emission derived from the distributions, and the values at the midplane (only for Experiments ii and iii),
3. the **Error Pass**, which, given the reference images and measurements, computes the mean square error (MSE) from the outputs of the forward pass,
4. and finally, the **Backward Pass**, where the gradients from the MSE objective are backpropagated to obtain the gradient of the objective with respect to the distributions as we are using a stochastic gradient descent optimisation method.

The gradient computed from 4., the Backward Pass, is then used to update the distributions using stochastic gradient descent which is detailed in 3.3.

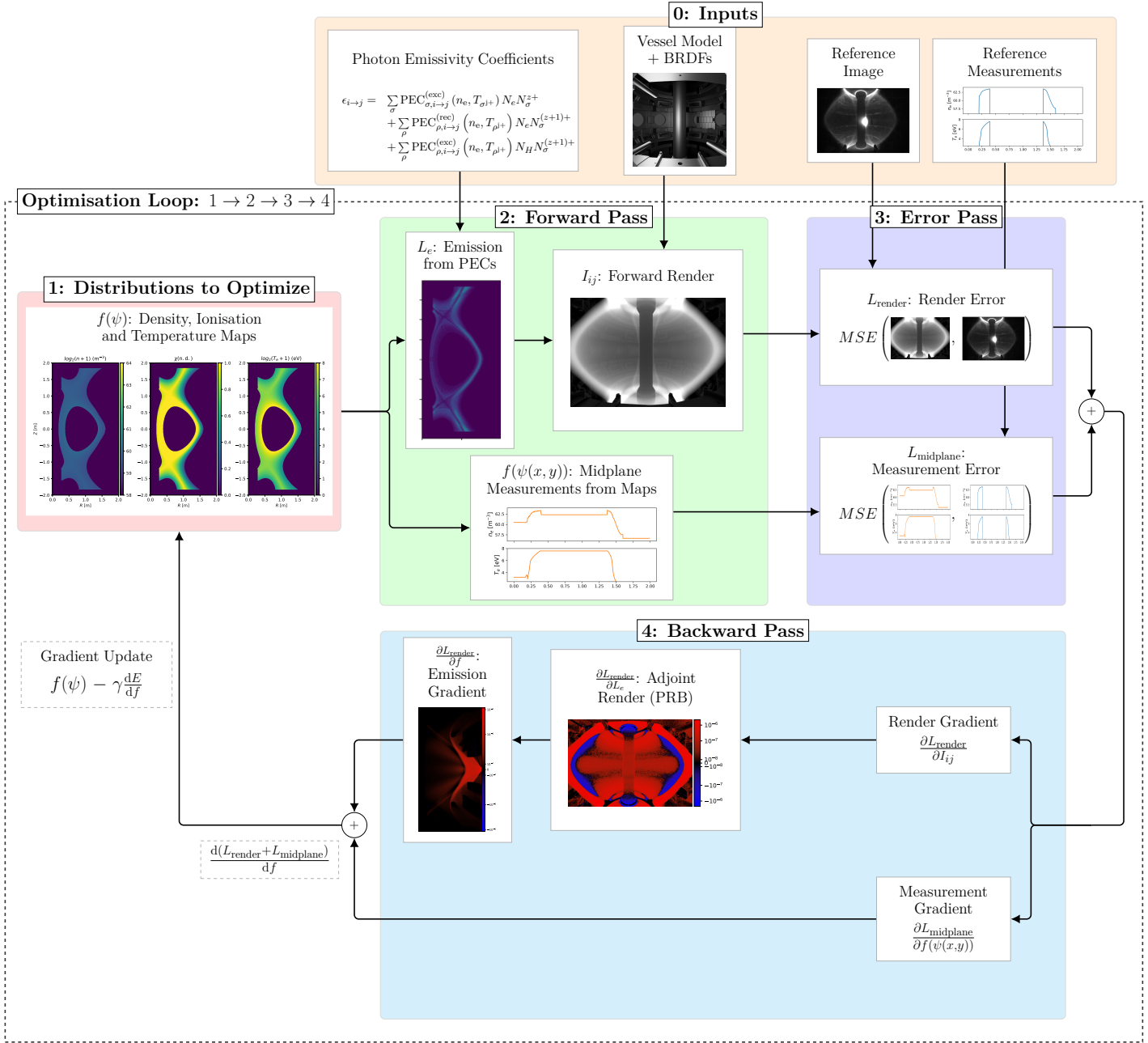


Figure 2: Simplified optimisation workflow diagram showing the main optimisation loop used for our reconstruction experiments. The top row shows the a-priori definitions required for optimisation. The middle row, from left to right, shows the maps to be optimised, the forward computation step, and the error calculation using the Mean Square Error (MSE). The bottom row, from right to left, shows the backward pass where first the gradients with respect to the render and the measurements are computed, then the adjoint render pass using Path-Replay Backpropagation and the gradients of the emission with respect to the inputs maps, before finally, they are combined to update the maps for optimisation.

2.2. The Light Emission and Transport Problem

The accurate simulation of the camera diagnostic can be split into two parts: Light Emission and Light Transport. The model used for estimating the plasma emission strongly influences the first-order radiance incident on the camera sensor while the light transport method determines the higher-order reflections.

2.2.1. Light Emission The plasma emission is determined by three non-equilibrium processes of excitation, recombination and charge-exchange which can be related to the total emission through the Photon Emissivity Coefficients (PECs) obtained from the Open-ADAS library [18]. The relationship between these three processes and the distributions of neutrals, electrons and temperature is given by

$$\begin{aligned} \epsilon_{i \rightarrow j} = & \sum_{\sigma} \text{PEC}_{\sigma, i \rightarrow j}^{(\text{exc})} (n_e, T_{\sigma^{j+}}) n_e N_{\sigma}^{z+} \\ & + \sum_{\rho} \text{PEC}_{\rho, i \rightarrow j}^{(\text{rec})} (n_e, T_{\rho^{j+}}) n_e N_{\sigma}^{(z+1)+} \\ & + \sum_{\rho} \text{PEC}_{\rho, i \rightarrow j}^{(\text{exc})} (n_e, T_{\rho^{j+}}) N_H N_{\sigma}^{(z+1)+} \end{aligned} \quad (1)$$

where n_e is the electron density, N_{σ}^{z+} is the density of species σ in the ionisation state $z+$, $T_{\sigma^{j+}}$ is the temperature of the ions of species σ in the ionisation state $j+$, and N_H is the density of thermally neutral hydrogen. The PECs are one of the Inputs shown in Figure 2.

For the purposes of this paper, we assume that, for a suitably filtered camera system, the emission from non-Deuterium sources can be ignored, and thus we can further simplify this equation. Assuming $n_{D+} \approx n_e$ imposed by the quasi-neutrality constraint, the fact that Deuterium has only one excited state, and that the PECs for charge-exchange are zero, 1 can be simplified to

$$\begin{aligned} \epsilon_{i \rightarrow j} = & \text{PEC}_{D,0 \rightarrow 1}^{(\text{exc})} (n_e, T_{D^{1+}}) n_e N_D^0 + \text{PEC}_{D,1 \rightarrow 0}^{(\text{rec})} (n_e, T_{D^{0+}}) n_e N_D^{1+} \\ = & \text{PEC}_{D,0 \rightarrow 1}^{(\text{exc})} (n_e, T_{D^{1+}}) n_e N_D^0 + \text{PEC}_{D,1 \rightarrow 0}^{(\text{rec})} (n_e, T_{D^{0+}}) n_e^2 \end{aligned} \quad (2)$$

While the PECs are defined for multiple spectral lines, we focus on D-alpha emission in line with a filtered camera system, though our pipeline permits incorporating additional spectral lines by simply summing the emission at each wavelength. Given that there is only a single spectral line, we will denote the total emission as $\epsilon_{D\alpha}$.

The PECs are provided in tabulated form as a function of electron density and electron temperature, thus, in order to optimise the density and temperature distributions through the emission, we interpolate the PECs using a locally differentiable bilinear interpolation scheme.

2.2.2. Light Transport Calculating the final image observed by the camera requires simulating the propagation of light within the vessel. This is the light transport problem for which we need an efficient method of tracing light paths that terminate at the camera sensor and originate at our light source, and summing them together such that we have an unbiased estimate. This can be solved via path-tracing methods which are Monte-Carlo methods for constructing light paths, starting from the sensor, and connecting them to the light source. Furthermore, as our light source is volumetric, we need to estimate the emission accrued along paths within a medium. And finally, this method must easily accommodate the calculation of gradients, ideally in an unbiased fashion as well.

Null-Scattering [12] is an ideal choice for estimating the light transport inside the plasma medium as it is unbiased and, unlike quadrature methods, scales efficiently with higher-order reflections. This is a technique that, inspired by Woodcock tracking [21] for the paths taken by neutrons in a dense medium, uses a homogenization of the medium and a simple probabilistic view of light interactions to compute the attenuation and transport of light. Considering a non-scattering spatially varying medium where the optical density is μ_t , and the emission is ϵ , null-scattering introduces a null density μ_n that homogenizes the medium such that $\bar{\mu}_t = \mu_t + \mu_n$ is constant throughout. This homogenized density is then used to sample free-flight distances into the medium using the Beer-Lambert law, $I_d/I_0 = \exp -d\bar{\mu}_t$ where d is the distance travelled into the medium and selecting, at each of these distances into the medium, an event with probability $\mu_n/\bar{\mu}_t$ whether or not the ray terminates. This is akin to simulating a single packet of light, sampling a distance into the medium and probabilistically choosing if it is absorbed. As these rays are traced from the camera back to the light source, whenever the ray interacts with the medium we accumulate the emission at the sampled point attenuated by the path taken by that ray. As we are looking at D-alpha emission and D-D plasmas are optically thin in D-alpha line, we take $\mu_t = 0$ and therefore $\bar{\mu}_t = \mu_n$. In order to simplify the optimisation problem and improve the efficiency of our method, we assume that deuterium is fully transparent. However, our method does not require this assumption.

Furthermore, as our plasma model has many parameters (e.g. 524 288 for one cross-section of resolution 512 by 1024) and the interior of the vessel is reflective leading to many interactions, we require a method of computing gradients that is memory efficient and mathematically correct. To that end, we use Path-Replay Backpropagation which leverages the reproducibility of pseudo-random numbers to "replay" the forward light transport and accumulate gradients at each interaction in a reverse-mode differentiation context, and unlike tape-based tracking of operations, has linear memory scaling in the number of parameters and reflections [20]. This is in contrast to naïve incorporation of automatic differentiation which would require not only storing the entire computational graph of the forward pass in memory, but then subsequently traversing this graph in reverse.

In the above discussion, we have abstracted how light is transported and mentioned

that it is being reflected by the interior surfaces of the vessel. In order to simulate these reflections we require a model of the vessel that describes not only the geometry, but also the way in which these surfaces scatter light. For the modelling of these surface reflections we use a Cook-Torrance Microfacet Bidirectional Reflectance Distribution Function (BRDF) [6] with a Beckmann Normal Distribution Function (NDF) [3]. We assume that all surfaces have a spatially homogeneous fixed albedo of 0.45 and a fixed roughness of 0.2. As for the geometry, we use the MAST CAD model, provided courtesy of the MAST Team, to model the interior surfaces, and we will be making the rendering compatible model available publicly. These correspond to the *Vessel Model+BRDFs* portion of the **Inputs** shown in Figure 2.

To compute the final image, the observed radiance incident on the camera sensor must be converted into an image as output by a real-world camera. While there are many characteristics inherent to this transformation, the most prominent of these in a calibrated camera system is the tonemapping, Region-of-Interest Cropping (ROI) and quantisation which we simulate as part of our rendering setup. We use an exposure and gamma-based tonemapping following

$$p_{ldr} = \begin{cases} (p_{hdr}\tau)^\gamma, & \text{for } p_{hdr} \cdot \tau \leq 1 \\ 1, & \text{for } p_{hdr} \cdot \tau > 1 \end{cases} \quad (3)$$

where p_{hdr} is the high-dynamic range pixel value, p_{ldr} is the tonemapped, low dynamic range pixel value, τ is the exposure time and γ is the gamma. We choose

$$\tau = 0.9Q(p_{hdr}, 97.5\%)^{-1} \quad (4)$$

where $Q(p_{hdr}, 97.5\%)$ is the 97.5-percentile of all the HDR pixel values.

2.3. Modelling Synthetic Plasma

In order to define the distributions that we optimise, corresponding to **1.** in Figure 2, we consider the representations that are suitable for optimisation. Similar to prior work, for the reconstruction of the neutrals density, we use toroidal Green's functions as the bases of our representation [4], but instead of emission we consider the neutrals distribution itself. In this representation, the toroidal bases are equivalent to the pixels of a 2D cross-section rotated toroidally about the central axis of the vessel.

For jointly representing the plasma distributions of n_e , n_n and T_e , the same representation, while sensible, does not adequately disambiguate line-of-sight emission around the X-points. In order to mitigate this limitation, we use a physically motivated parametric representation of the plasma composition and shape.

First we apply a change of variables from n_n , n_e and T_e to the total density (n_{total} [m^{-3}]), the ionisation fraction (χ [n.d.]) and the electron temperature (T_e [eV]) in order to enforce the inverse proportionality between neutrals and electron density. This is achieved with the forward transformation

$$n_{total} = n_n + n_e \quad (5a)$$

$$\chi = \frac{n_e}{n_n + n_e} \quad (5b)$$

$$T_e = T_e \quad (5c)$$

$$(5d)$$

and the inverse transformation,

$$n_n = (1 - \chi) n \quad (5e)$$

$$n_e = \chi n \quad (5f)$$

$$T_e = T_e \quad (5g)$$

between these two variables.

In order to enforce positivity of the outputs and better represent the high dynamic range of values, we encode the total density and the electron temperature following the transformation

$$x \mapsto \log_2(x + 1). \quad (5h)$$

Furthermore, in order to account for local variations in density, we split the total density into a bulk density component (n_{bulk}) and residual density component (n_{res}) such that

$$\log_2(n_{total} + 1) = n_{bulk} + n_{res} \quad (5i)$$

$$C_{residual} : (1/N) \sum_{i,j} |n_{res}^{i,j}| \leq \eta (1/N) \sum_{i,j} |n_{total}^{i,j}|, \eta < 1 \quad (5j)$$

where η controls the ratio between the magnitudes of the bulk and residual densities and $(1/N) \sum_{i,j}$ denotes the mean over the pixels of the cross-section. It should be noted that neither n_{bulk} nor n_{res} are strictly densities as they are summed in log-space to get n_{total} .

In a tokamak, the plasma shape is largely determined by the topology of the magnetic field, typically described by the so-called *flux surfaces*, contours of the poloidal magnetic potential $\psi(R, Z)$, where R and Z are the poloidal coordinates [2]. In order to exploit the shape information inherent in the magnetic field configuration, we model n_{bulk} , χ and T_e as functions of the plasma shape by defining them as tabulated piece-wise linear functions of the 2D normalised ψ_n -map. On the other hand, n_{res} is modelled as a 2D cross-section to allow for local deviations from the bulk density and decouple it from the magnetic field. Thus our full parametric representation of the maps is

$$n_{bulk} = n_{bulk}(\psi_n(R, Z)) \quad (6)$$

$$n_{res} = n_{res}(R, Z) \quad (7)$$

$$n_{total} = \exp_2[n_{bulk} + n_{res}] - 1 \quad (8)$$

$$T_e = T_e(\psi_n(R, Z)) \quad (9)$$

$$\chi = \chi(\psi_n(R, Z)). \quad (10)$$

2.4. Optimisation

We consider the reconstruction as an unconstrained optimisation problem in which our objective function is

$$\begin{aligned} L_{\text{total}} = & w_{\text{render}} L_{\text{render}} + w_{\text{cross-section}} L_{\text{cross-section}} + \\ & w_{\text{midplane}} L_{\text{midplane}} + w_{\text{TVA}} L_{\text{TVA}} + \\ & w_{\text{penalty}} L_{\text{penalty}} \end{aligned} \quad (11)$$

where L_{render} is the rendering error of the simulation of plasma emission and reflection, $L_{\text{cross-section}}$ is the error of the cross-sections themselves, L_{midplane} is the error of the values of the cross-sections at the midplane, L_{TVA} is the Total Variation Regularisation, L_{penalty} is the penalty-objective, and w_{render} , $w_{\text{cross-section}}$, w_{midplane} , w_{TVA} , w_{penalty} are the weights of each objective term.

We define the Mean Square Error as

$$\text{MSE}(x, \hat{x}) = (1/N) \sum_i (x_i - \hat{x}_i)^2 \quad (12)$$

and the Weighted Mean Square Error as

$$\text{WMSE}(x, \hat{x}, \zeta) = (1/N) \sum_i ((x_i - \hat{x}_i)/\zeta)^2 \quad (13)$$

allowing us to express the error terms in the above equation in a simpler form.

Rendering Error The rendering error is the mean square error between the reference image (img_{ref}) and the predicted image (img_{pred}),

$$L_{\text{render}} = \text{MSE}(\text{img}_{\text{ref}}, \text{img}_{\text{pred}}) \quad (14)$$

where only the under-exposed (pixels with values $\leq 1/255$) and over-exposed (pixels with values $\geq 1 - 1/255$) are ignored when computing the error. This corresponds to the Render Error in Figure 2.

Cross-Section Error The cross-section error is used for demonstrating the efficacy of the parametric representation and is the mean square error between the reference cross-sections (cs_{ref}) and the predicted cross-sections (cs_{pred}),

$$L_{\text{cross-section}} = \text{MSE}(\text{cs}_{\text{ref}}, \text{cs}_{\text{pred}}). \quad (15)$$

Midplane Error The midplane error is the mean square error between the reference (mid_{ref}) and the predicted (mid_{pred}) measurements at the midplane,

$$L_{\text{midplane}} = \text{WMSE}(\text{mid}_{\text{ref}}, \text{mid}_{\text{pred}}, \max(\text{mid}_{\text{ref}})) \quad (16)$$

normalized by the maximum value of the reference measurements. This corresponds to the Measurements Error in Figure 2.

Total Variation Error The Total Variation error,

$$L_{\text{TVA}} = \text{TVA}(\text{cs}_{\text{pred}}), \quad (17)$$

is computed following [15] and is used to smooth out very high-frequency ringing artifacts resulting from the noise in the predicted image and gradients

Penalty Objective The penalty-objective, L_{penalty} , arises from the fact that we have constraints on some of our variables while the Amsgrad optimiser is an unconstrained optimiser. The three constraints we have are $\chi \geq 0$, $\chi \leq 1$, and C_{residual} defined in 5j. In order to satisfy these constraints, we transform them in to an error term following the penalty-objective method [11]. Briefly, for an optimisation problem where the parameters are denoted by \vec{x} and the inequality constraint $g(\vec{x}) \leq 0$, we can introduce a functional P that takes a function $g(\vec{x})$ defining a constraint of the form $g(\vec{x}) \leq 0$ and transforms it into a penalty following

$$P[g] = \begin{cases} 0, & \text{for } g(\vec{x}) \leq 0 \\ g(\vec{x})^2, & \text{for } g(\vec{x}) > 0 \end{cases}. \quad (18)$$

Defining the functions g for each of our constraints, we can express them as

$$\chi \geq 0 \mapsto g_{\chi 0}(\chi) = -\chi \quad (19)$$

$$\chi \leq 1 \mapsto g_{\chi 1}(\chi) = \chi - 1 \quad (20)$$

$$C_{\text{residual}} \mapsto g_{\text{residual}} = (1/N) \sum_{i,j} |n_{\text{res}}^{i,j}| - \eta(1/N) \sum_{i,j} |n_{\text{total}}^{i,j}| \quad (21)$$

with C_{residual} defined in 5j. With this, we can then compute one penalty term for each of our constraints such that

$$L_{\text{penalty}} = P[g_{\chi 0}](\chi) + P[g_{\chi 1}](\chi) + P[g_{\text{residual}}](n_{\text{total}}, n_{\text{res}}). \quad (22)$$

3. Results

3.1. Experimental Setup

We implement the described differentiable rendering and optimisation pipeline using Mitsuba 3 [9] for the differentiable rendering and PyTorch [1] for the optimisation. We chose Mitsuba 3 as it can easily run on both CPUs and GPUs, and was the simplest to incorporate the chosen rendering algorithms into as it already has a mature set of tools for incorporating automatic differentiation into a rendering pipeline. PyTorch has a more mature set of optimisation tools so we opted to use it for most of the mathematical operations and the optimisation algorithms. All of our experiments are run on a single Nvidia A100-SXM4-80GB GPU on the CSD3 HPC Platform.

Given that many different optimisation problems can be considered within the same framework described above, we consider 3 different experiments of increasing difficulty where we look at reconstructing

- (i) the n_n distribution given known n_e and T_e distributions, demonstrated in 3.5,

- (ii) the distributions of n_n , n_e , and T_e , demonstrated in 3.6,
- (iii) the distributions with an ROI cropped sensor image and quantisation of the image, demonstrated in 3.7.

We use an experimentally validated SOLPS simulation case on MAST shot 30356 [8] to validate our framework as it provides a physical ground truth that we can compare to irrespective of measurement noise, and the SOLPS simulation we consider includes the neutral Deuterium density distribution which is not part of the diagnostic data measured during experiments. Additionally, the camera data we have access to, at the time of writing, lacks metadata for the exposure and sensor response function, as well as necessary calibrations of the sensor noise characteristics and the per-pixel black levels which are required to use the data in a computer vision application such as ours.

As the SOLPS simulation uses a mesh with spatially varying resolution, we bi-linearly interpolate these values onto fixed resolution maps with a resolution of 512 by 1024. We use a segmentation map to identify the pixels in the fixed resolution maps that correspond to the SOLPS mesh, and render and optimise only the pixels corresponding to the mesh as we do not have any priors for regions outside the mesh.

The first row in Figure 3 shows the cross-sections obtained from the SOLPS simulation and we note that this simulation extends only partially into the plasma core. While we expect the emission in the core to be low in actual experiments due to the sparsity of neutrals, any emission from this region would be included in high-speed camera images and may provide insight into the composition in the core. Similarly, Thomson Scattering measurements also include measurements from the core which we lack in our mid-plane profiles as they are absent from the SOLPS mesh. The second row shows the capability of our parametric representation and is a validation that this is a useful way of modelling synthetic plasma.

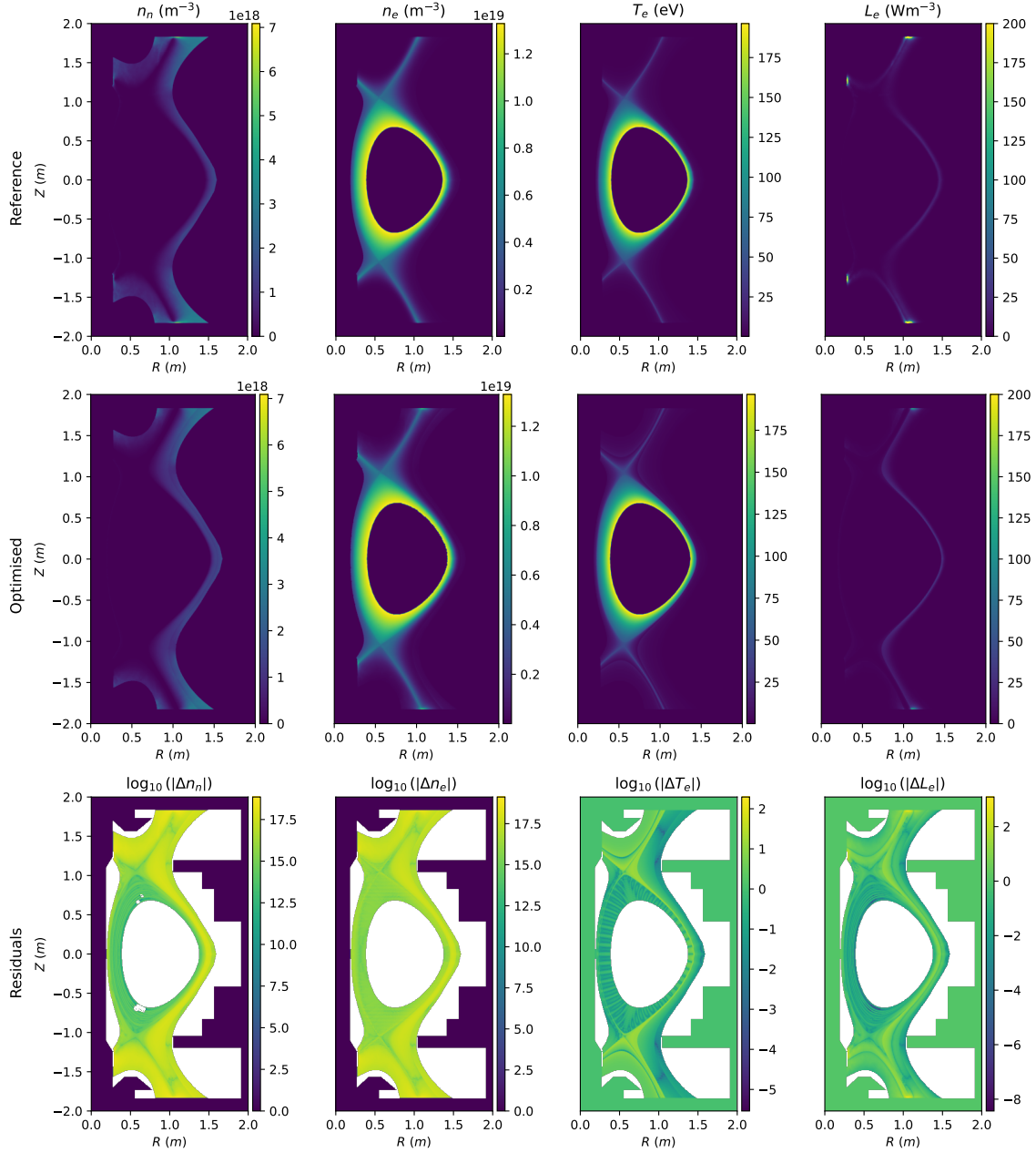
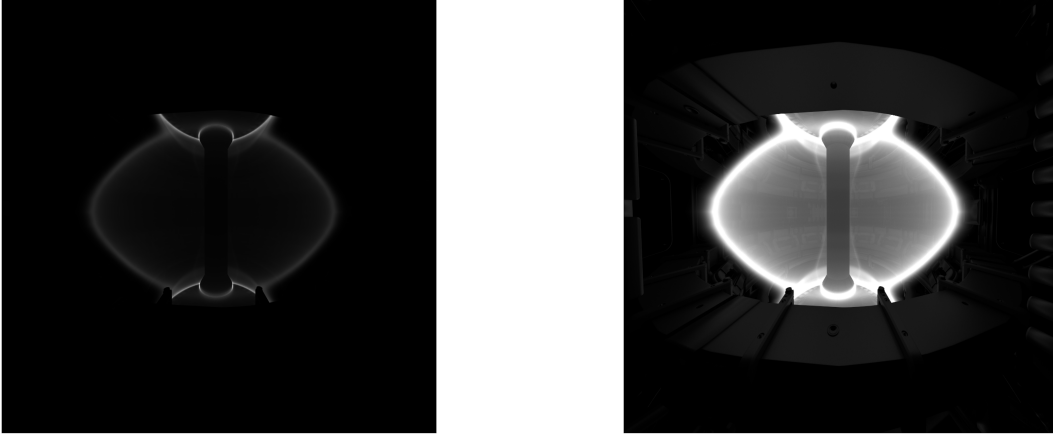


Figure 3: First Row: Cross-sections obtained from a SOLPS simulation of MAST shot 30356 [8] as interpolated onto a regular 2D grid. Second Row: Parametric representation of the cross-sections using the formulation described in 2.3. Third Row: Residuals of the parametric reconstruction. The absolute \log_{10} difference between the maps in the first row and the second row.

Figure 4 shows the rendering of the plasma parameters from SOLPS using 1024 samples per pixel which takes approximately 6 minutes to render. The camera position and focal length has been calibrated using the CALCAM software [16] and the MAST CAD model, following the same procedure as used in Farley et al. [7] and the parameters of which are in Appendix A. Using the tonemapping in 3, we set the exposure to 0.109s



(a) High-dynamic range render scaled to range $[0, 1]$.

(b) Tonemapped render using an exposure time of 0.109s and a gamma of 0.625.

Figure 4: Rendered using Mitsuba 3 [9] from SOLPS cross-sections using PEC coefficients obtained from Open-ADAS [18] to compute the volumetric emission.

which ensures that most of the pixels are well-exposed, and the gamma to 0.625 as this qualitatively matches the real images seen in Figure 1. For Experiments i and ii, the exposure is set to 0.109s and for Experiment iii, the exposure is set to 0.0794s as there are proportionally more bright pixels than dark.

Qualitatively, these renders closely resemble the images shown in Figure 1 in terms of the overall distribution of intensities, the appearance of reflections, and the primary regions of emission. They differ in that the high-speed camera used to capture the images in Figure 1 is also subject to systematic and random noise which our rendering pipeline excludes.

3.2. Reconstructing Cross-Sections

While directly optimising the cross-sections at the full resolution of 512 by 1024 with high-resolution tabulated functions is possible, the high-dimensionality of the maps increases the complexity of the optimisation and does not exploit the spatial correlations between neighbouring points in our tabulated representation. To exploit these correlations and improve the computational efficiency of our method, we consider a hierarchical approach where, instead of using our full-resolution distributions as the first stage in Figure 2, we start with a lower resolution representation of 32 by 64 cross-sections and increase the resolution by a factor of four using bilinear interpolation each time the error plateaus. Similarly, our tabulated functions start with a 64 segment representation and are linearly upsampled by a factor of 4 as well. This is repeated 3 times to achieve a final resolution of 512 by 1024 in the cross-section and 1024 segments

in our tabulated functions.

For Experiment i, as we consider only a single cross-section representing the neutrals density, we start with a n_n cross-section of resolution 32 by 64 with an initial uniform density of 10^{17} as this avoids both significant over- and under-exposure. As stated before, this is then upscaled over three optimisation passes to reach a final resolution of 512 by 1024.

For Experiments ii and iii, we start with a n_{res} -map with resolution 32 by 64, and a tabulated function of n_{bulk} , χ and T_e with 64 segments ranging from $\psi_n = 0.769$ to $\psi_n = 1.42$ which are the minimum and maximum values of ψ_n on the SOLPS mesh. The n_{res} -map and the tabulated functions are then up-scaled by a factor of 4 using bilinear and linear scaling respectively 3 times to achieve a final resolution of 512 by 1024 and 1024 segments respectively. Each of our maps are initialized as follows:

- (i) the n_{res} -map is initialised to a constant value of 1 to avoid zero gradients,
- (ii) the n_{bulk} -function is initialised to a linearly decaying value corresponding to $\log_2(n_{bulk}^{(core)} + 1) = 20$ at $\psi_n = 0$ to $\log_2(n_{bulk}^{(SOL)} + 1) = 19$ at $\psi_n = 1$, i.e. the initial function is defined as

$$\log_2(n_{bulk} + 1) = \log_2(n_{bulk}^{(core)} + 1) + \psi_n \cdot \log_2 \left(\frac{n_{bulk}^{(SOL)} + 1}{n_{bulk}^{(core)} + 1} \right), \quad (23)$$

- (iii) the T_e -function to a linearly decaying value corresponding to $\log_2(T_e^{(core)} + 1) = 2 \cdot 10^4$ at $\psi_n = 0$ to $\log_2(T_e^{(SOL)} + 1) = 2 \cdot 10^1$ at $\psi_n = 1$, i.e. the initial function is defined as

$$\log_2(T_e + 1) = \log_2(T_e^{(core)} + 1) + \psi_n \cdot \log_2 \left(\frac{T_e^{(SOL)} + 1}{T_e^{(core)} + 1} \right), \quad (24)$$

- (iv) the χ -function to a linearly decaying value corresponding to $\chi^{(SOL)} = 1$ at $\psi_n = 1$ to $\chi^{(outer)} = 0$ at $\psi_n = 2$, i.e.

$$\chi = \begin{cases} 1.0, & \text{for } \psi_n \leq 1 \\ \chi^{(SOL)} + (\psi_n - 1.0) \cdot (\chi^{(outer)} - \chi^{(SOL)}), & \text{for } \psi_n > 1 \end{cases}. \quad (25)$$

These initialisations are chosen based on a rough expectation of the distribution of the plasma where we expect that within the core the plasma is fully ionised and dense, and outside it is less dense and with a lower ionisation as we move further from the plasma edge, and, similarly, we expect that at the very center of the plasma core, the temperature is at its highest and decaying until the edge where there is a significant drop (the pedestal in high-confinement mode).

3.3. Optimisation Setup

The optimisation framework discussed in Section 2.4 permits many ways of selecting the initial step size — γ in Figure 2, how many optimisation steps to take, how to vary the step size over the course of the optimisation, and when to terminate the optimisation. We use the default step size of Amsgrad, 10^{-3} in Experiments ii and

iii, and a step size of 1.0 in Experiment i. We increase the number of optimisation steps at each up-scaling stage, roughly quadrupling the number of optimisation steps each time. Furthermore, rather than simply run the optimisation for a large number of fixed steps, we systematically reduce the step size using PyTorch’s ‘ReduceLROnPlateau’ [1] which tracks the improvement of the error and reduces it whenever it does not improve for ‘patience’ number of steps. The ‘patience’ is then increased linearly during each optimisation pass as higher resolution maps require more steps to achieve comparable improvement, and we terminate the optimisation whenever reducing the step size by a factor of 10^2 stops improving the error.

As the rendering is subject to noise, L_{render} is a stochastic estimate of the true error and similarly, as the backward render pass is noisy, the gradients are also a stochastic estimate of the true gradient. In order to account for the noise, the samples per pixel at each iteration of the optimisation is increased monotonically following $\lfloor 2.25 \cdot L_{\text{render}}^{-0.5} \rfloor$. This is roughly the variance scaling of the underlying rendering and ensures that the noise floor in the render and gradients decreases along with the error itself, thus allowing us to resolve both the render error and the gradients with accuracy proportional to the error itself. This also has the advantage of accelerating the optimisation as we avoid using high samples per pixel in the early stages of the optimisation. Additionally, as the sampling rate of the emission is proportional to μ_n and thus increasing this gives a more accurate estimate of the emission for each ray that passes through the medium, we set its value to 2^{u+2} where u is the optimisation pass ranging from 0 to 2 for the 3 optimisation passes.

3.4. Objective Functions

Depending on the reconstruction experiment, we use a different weighting of each of the terms in 11.

For Experiment i, we set $w_{\text{midplane}} = 0$, $w_{\text{cross-section}} = 0$ and $w_{\text{penalty}} = 0$ as we do not have mid-plane neutrals measurements, we are attempting to optimise through the rendering itself and so we do not explicitly enforce a match to the reference cross-sections, and the encoding in $5h$ explicitly enforces the positivity of the cross-section. Furthermore, we do not mask out pixels for this experiment as the under- and over-exposed pixels force the estimated n_n in these regions to attain a lower bound and an upper bound respectively of the true neutrals density. As we do not want to overly smooth the optimised cross-section, we set $w_{\text{TVA}} = 10^{-8}$.

For Experiments ii and iii, we set $w_{\text{cross-section}} = 0$ term as, again, we want to reconstruct the cross-sections without explicitly enforcing a match to the reference cross-sections. As we do not want to overly smooth the optimised cross-section, we set $w_{\text{TVA}} = 10^{-3}$.

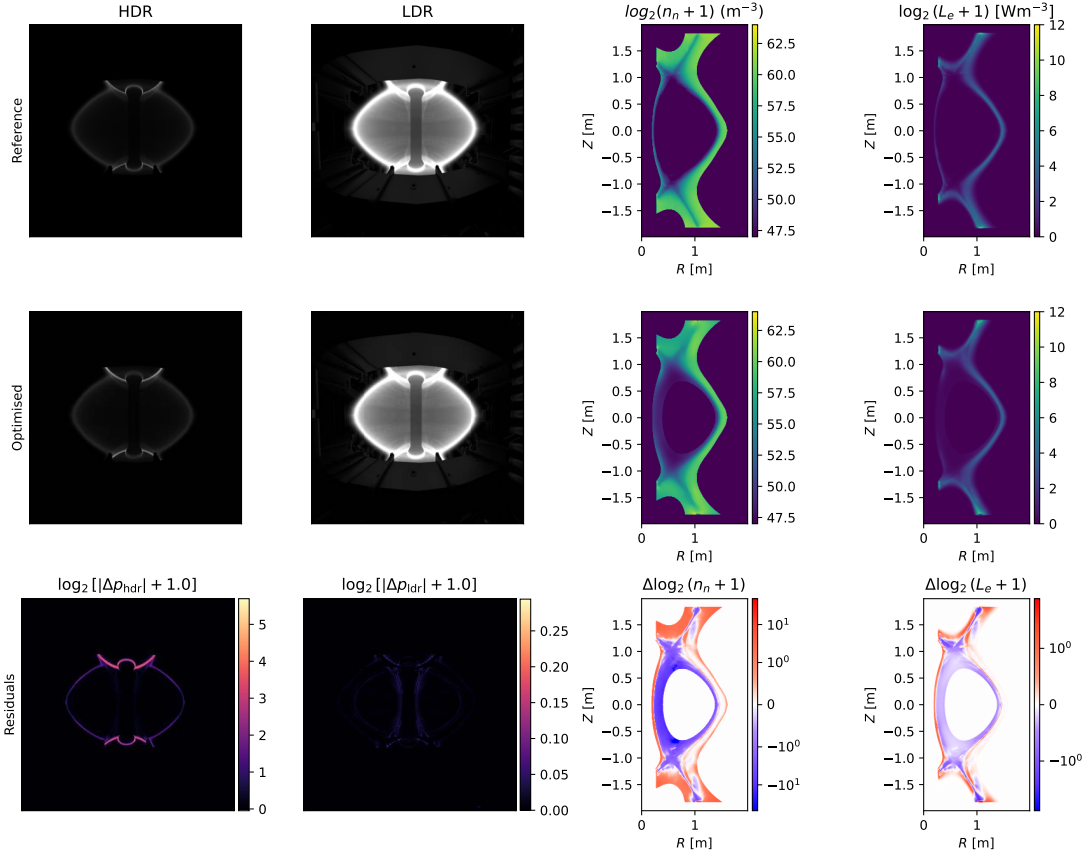


Figure 5: Experiment i: Neutrals reconstruction using n_e and T_e as known priors. The first row shows the reference images and cross-sections, the second row shows the reconstructions and the last row shows the residuals of these reconstructions. The renders are tonemapped using $\tau = 0.109s$ and $\gamma = 0.625$.

3.5. Experiment i: Reconstructing the Neutrals Density

The neutrals density reconstruction, shown in Figure 5, takes 21 hours to converge, and achieves a rendering error of $1.58 \cdot 10^{-5}$ and a mean square error on the log-encoded cross-section, $\log_2(n_n + 1)$, of 5.97 compared to an error of $1.95 \cdot 10^1$ for the initial value of $n_n = 10^{17}$. The mean square error on the underlying n_n map is $1.07 \cdot 10^{35}$.

We see that the differentiable rendering routine achieves a plausible reconstruction of the neutrals density where the core neutrals is low and the density in the rest of the vessel is reasonably distributed. In comparison to the reference cross-section, this reconstruction captures the fine features of the legs, the location of the LCFS and the distribution of densities around the X-Points. It is also able to accurately reproduce the locally high-emission at the divertor strikes that results from the reflection of neutrals from the vessel walls.

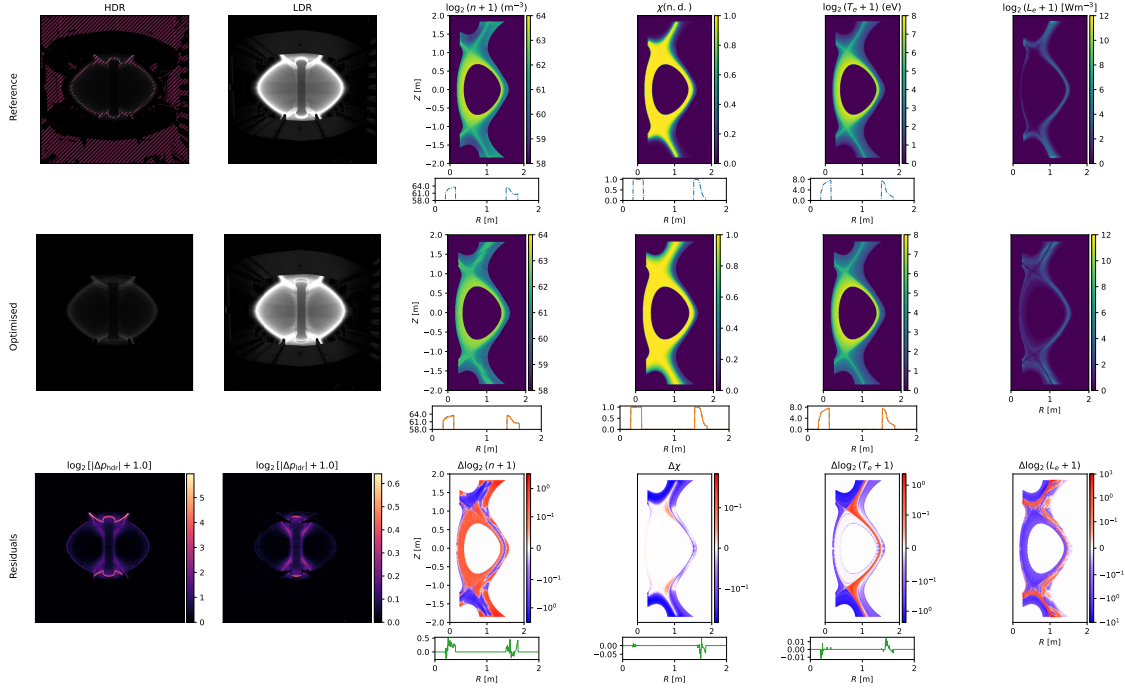


Figure 6: Experiment ii: Reconstruction of total density n , ionisation fraction χ and electron temperature T_e using renders and midplane values as the ground truth. The first row shows the reference images and cross-sections while the second row shows the reconstructions. The renders are tonemapped using $\tau = 0.109$ s and $\gamma = 0.625$.

3.6. Experiment ii: Reconstructing n_n , n_e , and T_e using the Full-Sensor

In Figure 6, we show the results of reconstructing the distributions of n_n , n_e , and T_e given the full sensor image. This optimisation took roughly 26 hours and we see that it is able to achieve a mean square rendering error of $8.30 \cdot 10^{-4}$ on the tonemapped, low dynamic range image, and a mean square error of 1.53 on the high dynamic range image. This translates to a very close match to the ground truth low dynamics range image, but, as can be seen at the divertor strike points, differs significantly due to mismatch in intensity. Part of this mismatch is due to the fact that we filter out pixels that are over-exposed and thus the optimisation receives no signal from these pixels, but another part is due to the fact that these pixels are clamped to a value of 1 despite being significantly brighter leading to the higher error seen in the high dynamic range image.

We see that not only does the render match the ground truth image, but the underlying maps also accurately reconstruct the ground truth maps. The midplane profiles of all three maps closely match the ground truth and we see that this is translated well off-midplane. Furthermore, the density is well reconstructed across the entire map, with the expected shape and capturing some of the local variation in density. While there are still some high frequency artifacts along the flux surfaces, this is due to the stochasticity of the optimisation and could be further smoothed out with postprocessing. Furthermore, we report the exact errors achieved on these maps using two different

metrics in Appendix B.

3.7. Experiment iii: Reconstructing with a Cropped and Quantised Image

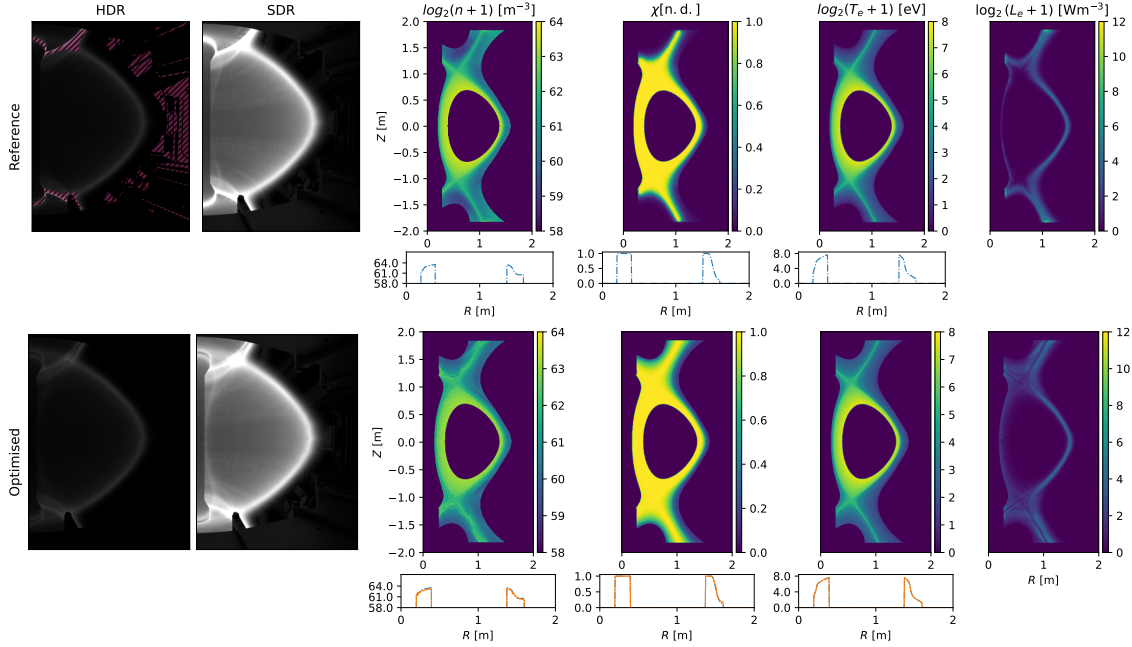


Figure 7: Experiment iii: Reconstruction of total density n , ionisation fraction χ and electron temperature T_e using cropped and quantised renders, and midplane values as the ground truth. The first row shows the reference images and cross-sections while the second row shows the reconstructions. The renders are tonemapped using $\tau = 0.0794s$ and $\gamma = 0.625$.

In Figure 7, we show the results of Experiment iii where the render is cropped and the reference image is 8-bit quantised. With fewer pixels per render, this optimisation took 5 hours to converge. The rendering mean square error achieved by this optimisation is $1.71 \cdot 10^{-3}$ for the low dynamic range image and 3.49 for the high dynamic range image where we observe a similar discrepancy at the divertor strikes points.

We see that this optimisation achieves comparable results to the reconstruction with the full sensor despite a) running faster due to the reduced number of pixels in the rendering, and b) the presence of quantisation effects on the pixel values. The biggest difference compared to the full sensor is the increased noise in the density distributions.

4. Robustness of the Optimisation

In order to explore the robustness of the optimisation, we first consider varying the initialisation in Experiment iii, and then discuss the remaining sources of error that could be contributing to the discrepancy between the references and our optimised results.

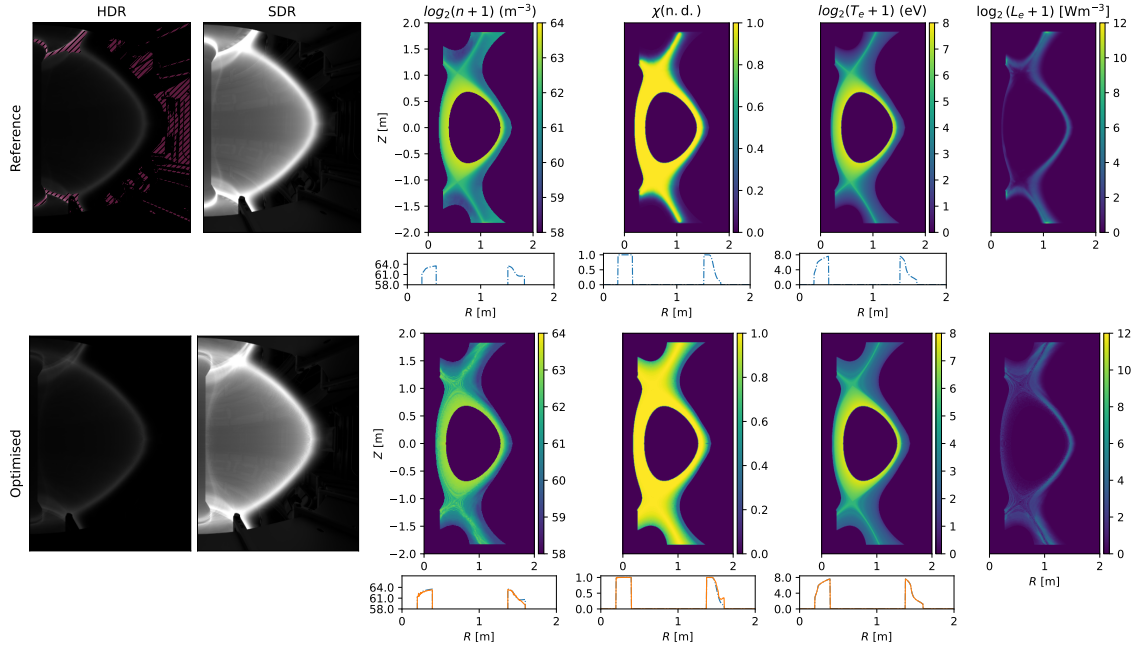


Figure 8: Reconstruction following same setup as Experiment iii with the ionisation fraction initialised to 0.5 at the mesh boundary. The first row shows the reference images and cross-sections while the second row shows the reconstructions. The renders are tonemapped using $\tau = 0.0794s$ and $\gamma = 0.625$.

4.1. Varying the Initialisation

In 3.6 and 3.7, we demonstrated successful reconstruction of the underlying maps starting from a specific initialisation of the maps and tabulated functions. Here we show the robustness of the method with regards to these initialisations, specifically in varying the ionisation fraction, χ .

We choose χ to vary as the emission is proportional to the product $n_n \cdot n_e$ and thus decreasing n_n can, locally, be compensated by a proportional increase in n_e . We demonstrate this many-to-one mapping issue using the same setup as the ROI cropped and quantised optimisation, except we set χ to 0.5 at the extremes of the SOLPS mesh instead of 0.

We can see that despite varying the initialisation, the optimisation is able to converge to a valid result that visually reproduces the input image and captures the global distributions, but is unable to accurately determine the total density and ionisation fraction far outside the SOL. In comparison to Experiment iii, we achieve a mean square error of $1.31 \cdot 10^{-3}$ on the tonemapped render, and 3.50 on the HDR render, but the optimisation is unable to discern the true distribution of neutrals density, and under-estimates the neutrals distribution outside of the LCFS as indicated by the higher than ground truth ionisation fraction.

4.2. Other Sources of Error

As mentioned before, we assume that the ionisation fraction can be modelled as a function of the normalised magnetic flux, but, from Figure 3, we see that the residual in the neutrals and electron densities outside the core exhibit a jump across the SOL which indicates that this assumption does not hold as strongly outside the core. Succinctly, the assumption that n_{bulk} , χ and T_e can be modelled as functions of ψ_n is violated proportional the deviation of ψ_n from 1. This also manifests in the divertor regions where both the n_n and n_e distributions are accurate only in a narrow region around the plasma legs where $\psi_n = 1$.

While representation plays a role in the accuracy of the underlying reconstructions, the emission is also impacted by the variation in electron density and electron temperature. When the electron density and temperature are both outside the region for which $D\alpha$ -emission is peaked, the corresponding region is either faint or entirely invisible in the resulting render. Furthermore, as can be seen in Eq. 2, the emission is linearly proportional to both n_n and n_e for a fixed temperature (i.e. fixed PEC) and thus the imaging signal only provides information about their product, $n_n n_e$.

Finally, we observe that, due to the scaling of the variance in the render, reducing the noise floor by an order of magnitude requires us to increase the samples per pixel by more than a factor of 10. We account for this variation by considering a scaling of the samples in proportion to the achieved error, but this creates a lower bound for the achievable error given a fixed computational budget.

5. Conclusion

In this paper, we have demonstrated that differentiable rendering can serve as a valuable tool for inverse estimation of plasma parameters from image data. While previous works exist in the form of rendering and least-squares fitting to image measurements, our technique enables end-to-end rendering and reconstruction with better computational scaling, and easy integration of multi-modal measurements in the form of midplane measurements. Although we have motivated our method with toroidally symmetric plasma distributions and purely $D\alpha$ -line emission for reconstruction, the rendering pipeline we have described is not restricted to specific representations nor to $D\alpha$ -line emission, the exploration of which, such as 3d filamentary structures, is left as future work.

Our contributions include the reconstruction of the neutrals density distribution which, to our knowledge, almost always required expensive Monte-Carlo simulations to reconstruct prior to this technique. As our method recovers plausible distributions of neutrals density from imaging and measurements, our reconstructions can complement SOLPS simulations with neutrals transport by providing validation and comparison to experiments, and perhaps even providing a better prior for the simulations at a relatively modest computational cost. While we have reported timings of the optimisation and

rendering on a single GPU, the algorithm itself scales trivially and thus could be parallelised easily across multiple GPUs leading to further speed-ups in the process.

The reconstruction pipeline demonstrated in our work also motivates the use of calibrated and filtered imaging systems on the MAST-U tokamak in the near future as this would enable new 2D diagnostic mapping of various plasma quantities, including electron density and temperature distributions, neutrals distribution, and eventually impurity distributions. High-Dynamic Range imaging could be used to avoid pixel under-/over-exposure that limits the information that can be extracted from the strike points. As proposed by Carr et al. [5], the BRDF of the vessel walls could be regularly measured and used as part of the rendering pipeline for estimated properties of the plasma. Furthermore, as our rendering pipeline is agnostic to how the emission is computed, additional diagnostics such as bolometry, and physical models such as simulations themselves can be used to constrain the space of possible distributions.

In Section 4 we considered the most likely contributors to the final error of the optimisations, but the lack of ground truth data such as more SOLPS+EIRENE cases limits our ability to disentangle each of their contributions. Furthermore, due to the difficulty of inverting many-to-one mappings, we expect that future work will include statistical modelling of likely inversions based on the input image and measurements. A possible direction is to train a neural network on a dataset of pairs of images and midplane measurements, and estimate the underlying distributions of n_n , n_e , and T_e .

We have also shown that the availability of gradients greatly improves the convergence and stability of non-linear optimisation based inverse estimation. As auto-differentiation is not only limited to light transport simulations, we hope that our work will encourage the adoption of differentiable codes in the wider physics community.

Acknowledgements

We would like to thank David Moulton for sharing with us both the SOLPS data and the code for reading it. This work was partly supported by the UK Atomic Energy Authority (Agreement Number 12038). Pieter Peers was supported in part by NSF grant IIS-1909028 and DOE grant DE-SC0024624. This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion) and from the EPSRC [grant number EP/T012250/1]. To obtain further information on the data and models underlying this paper please contact PublicationsManager@ukaea.uk*. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them. This work was performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service (www.csd3.cam.ac.uk), provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research

Council (capital grant EP/T022159/1), and DiRAC funding from the Science and Technology Facilities Council (www.dirac.ac.uk).

References

- [1] Ansel J et al. “PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation”. In: *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM, Apr. 2024. DOI: 10.1145/3620665.3640366.
- [2] Appel L and Lupelli I. “Equilibrium reconstruction in an iron core tokamak using a deterministic magnetisation model”. In: *Computer Physics Communications* 223 (Feb. 2018), pp. 1–17. DOI: 10.1016/j.cpc.2017.09.016.
- [3] Beckmann P and Spizzichino A. *The Scattering of Electromagnetic Waves from Rough Surfaces*. A Pergamon Press book. Pergamon Press; [distributed in the Western Hemisphere by Macmillan, New York], 1963. ISBN: 9780080100074.
- [4] Carr M et al. “Description of complex viewing geometries of fusion tomography diagnostics by ray-tracing”. In: *Review of Scientific Instruments* 89.8 (Aug. 2018), p. 083506. ISSN: 0034-6748. DOI: 10.1063/1.5031087.
- [5] Carr M et al. “Physically principled reflection models applied to filtered camera imaging inversions in metal walled fusion machines”. In: *Review of Scientific Instruments* 90.4 (4 Apr. 2019), p. 043504. ISSN: 10897623. DOI: 10.1063/1.5092781.
- [6] Cook RL and Torrance KE. “A Reflectance Model for Computer Graphics”. In: *ACM Trans. Graph.* 1.1 (Jan. 1982), pp. 7–24. ISSN: 0730-0301. DOI: 10.1145/357290.357293.
- [7] Farley T et al. “Filament identification in wide-angle high speed imaging of the mega amp spherical tokamak”. In: *Review of Scientific Instruments* 90.9 (Sept. 2019), p. 093502. ISSN: 0034-6748. DOI: 10.1063/1.5109470.
- [8] Havlíčková E et al. “SOLPS analysis of the MAST-U divertor with the effect of heating power and pumping on the access to detachment in the Super-x configuration”. In: *Plasma Physics and Controlled Fusion* 57.11 (Aug. 2015), p. 115001. ISSN: 1361-6587. DOI: 10.1088/0741-3335/57/11/115001.
- [9] Jakob W et al. *Mitsuba 3 renderer*. Version 3.1.1. <https://mitsuba-renderer.org>. 2022.
- [10] Kantarci MG, Gokberk B, and Akarun L. “A Survey of 3D Object Reconstruction Methods”. In: *2022 30th Signal Processing and Communications Applications Conference (SIU)*. IEEE, May 2022. DOI: 10.1109/siu55565.2022.9864922.
- [11] Luenberger DG and Ye Y. “Penalty and Barrier Methods”. In: *Linear and Nonlinear Programming*. Springer International Publishing, 2016, pp. 397–428. ISBN: 9783319188423. DOI: 10.1007/978-3-319-18842-3_13.

- [12] Miller B, Georgiev I, and Jarosz W. “A null-scattering path integral formulation of light transport”. In: *ACM Transactions on Graphics* 38.4 (July 2019), pp. 1–13. DOI: 10.1145/3306346.3323025.
- [13] Nimier-David M et al. “Unbiased inverse volume rendering with differential trackers”. In: *ACM Trans. Graph.* 41.4 (July 2022). ISSN: 0730-0301. DOI: 10.1145/3528223.3530073.
- [14] Niu Y et al. “Overview of image-based 3D reconstruction technology”. In: *Journal of the European Optical Society-Rapid Publications* 20.1 (2024), p. 18. ISSN: 1990-2573. DOI: 10.1051/jeos/2024018.
- [15] Rodríguez P. “Total Variation Regularization Algorithms for Images Corrupted with Different Noise Models: A Review”. In: *Journal of Electrical and Computer Engineering* 2013.1 (2013), p. 217021. DOI: <https://doi.org/10.1155/2013/217021>.
- [16] Silburn S et al. *Calcam*. en. 2024. DOI: 10.5281/ZENODO.10655746.
- [17] Tong X. “Inverse Rendering”. In: *Computer Vision*. Springer International Publishing, 2021, pp. 711–713. ISBN: 9783030634162. DOI: 10.1007/978-3-030-63416-2_804.
- [18] University of Strathclyde. *Open-ADAS*. Version 2.1. Retrieved on 30/06/2021. The ADAS Project. 2021. URL: <https://open.adas.ac.uk/>.
- [19] Vicini D, Speierer S, and Jakob W. “Differentiable Signed Distance Function Rendering”. In: *Transactions on Graphics (Proceedings of SIGGRAPH)* 41.4 (July 2022), 125:1–125:18. DOI: 10.1145/3528223.3530139.
- [20] Vicini D, Speierer S, and Jakob W. “Path Replay Backpropagation: Differentiating Light Paths using Constant Memory and Linear Time”. In: *Transactions on Graphics (Proceedings of SIGGRAPH)* 40.4 (Aug. 2021), 108:1–108:14. DOI: 10.1145/3450626.3459804.
- [21] Woodcock E et al. “Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry”. In: *Proc. Conf. Applications of Computing Methods to Reactor Problems*. Vol. 557. 2. Argonne National Laboratory. 1965.

Appendix A. Camera Calibration

The camera calibration is done through the use of the MAST CAD model and the CALCAM software. In order to calibrate the image and correctly estimate the distortion parameters, we first pad the incoming image with zeros on each side based on the sensor pixel width/height and the Region-of-Interest (ROI) crop that was used. Without this step the camera position and orientation would be incorrectly estimated in relation to a synthetic camera using the same ROI. This also enables us to simulate different ROI crops without having to recalibrate the camera position and orientation. Figure A1

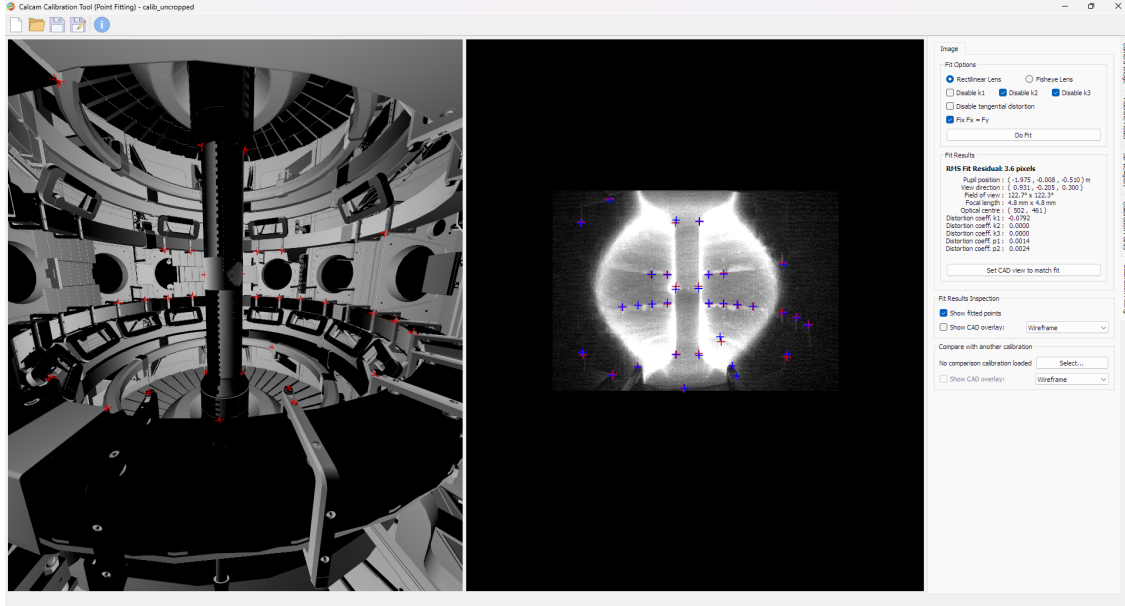


Figure A1: Camera calibration showing the MAST CAD alongside a frame from shot 30301 in CALCAM. The camera image has been brightened and the contrast decreased in order to bring out the visibility of the vessel interior.

shows a screenshot from the CALCAM software showing the number of points used for calibration and Eq. A.1 shows the calibrated camera parameters.

$$(p_x, p_y, p_z) = (-1.975, -0.008, -0.510) \text{ m} \quad (\text{A.1})$$

$$(d_x, d_y, d_z) = (0.931, -0.205, 0.300) \text{ m} \quad (\text{A.2})$$

$$(f_x, f_y) = (4.8, 4.8) \text{ mm} \quad (\text{A.3})$$

$$(o_x, o_y) = (502, 461) \text{ px} \quad (\text{A.4})$$

$$(k_1, k_2, k_3) = (-0.0792, 0.0, 0.0) \quad (\text{A.5})$$

$$(p_1, p_2) = (0.0014, 0.0024) \quad (\text{A.6})$$

$$(\text{A.7})$$

In this paper we do not utilise the distortion parameters nor the offset optical center, and are only interested in the camera extrinsics and the focal length. We also note that the camera moves during and between shots and therefore these calibrations would be updated for optimisation on real images.

Appendix B. Cross-Sections Errors from Experiments

Here we report the exact errors achieved by the experiments on the various underlying maps and representations. Our metrics are the mean square error and the mean absolute error. In B1 we show the errors achieved on Experiment ii, the full sensor reconstruction, in B2 we show the errors achieved on Experiment iii, the ROI cropped

Table B1: Errors achieved by the optimisation of the parametric maps using the renders and midplane measurements, both in the parametric representation and in the SOLPS representation. MSE is the Mean Square Error and MAE is the Mean Absolute Error.

	Errors on Experiment ii			
	MSE		MAE	
	Whole Map	Midplane	Whole Map	Midplane
$\log_2 (n_{total} + 1)$	$8.89 \cdot 10^{-2}$	$1.00 \cdot 10^{-2}$	$1.23 \cdot 10^{-1}$	$3.83 \cdot 10^{-2}$
χ	$1.69 \cdot 10^{-2}$	$5.62 \cdot 10^{-5}$	$3.63 \cdot 10^{-2}$	$1.76 \cdot 10^{-3}$
$\log_2 (T_e + 1)$	$7.92 \cdot 10^{-2}$	$3.24 \cdot 10^{-6}$	$9.17 \cdot 10^{-2}$	$4.74 \cdot 10^{-4}$
n_n	$9.60 \cdot 10^{34}$	$2.18 \cdot 10^{33}$	$7.94 \cdot 10^{16}$	$9.05 \cdot 10^{15}$
n_e	$6.71 \cdot 10^{35}$	$3.19 \cdot 10^{35}$	$3.22 \cdot 10^{17}$	$1.94 \cdot 10^{17}$
T_e	$6.62 \cdot 10^0$	$9.95 \cdot 10^{-4}$	$7.15 \cdot 10^{-1}$	$7.02 \cdot 10^{-3}$
$\epsilon_{D\alpha}$	$4.39 \cdot 10^{-1}$		$2.32 \cdot 10^{-1}$	

and quantised image reconstruction, and in B3 we show the errors achieved on the modified Experiment iii where the ionisation fraction, χ , is initialized to a higher value of 0.5 on the SOLPS mesh boundary.

Table B2: Errors achieved by the optimisation of the parametric maps using the renders and midplane measurements, both in the parametric representation and in the SOLPS representation. MSE is the Mean Square Error and MAE is the Mean Absolute Error.

	Errors on Experiment iii			
	MSE		MAE	
	Whole Map	Midplane	Whole Map	Midplane
$\log_2 (n_{total} + 1)$	$9.15 \cdot 10^{-2}$	$1.35 \cdot 10^{-2}$	$1.28 \cdot 10^{-1}$	$4.56 \cdot 10^{-2}$
χ	$1.71 \cdot 10^{-2}$	$9.18 \cdot 10^{-5}$	$3.70 \cdot 10^{-2}$	$2.37 \cdot 10^{-3}$
$\log_2 (T_e + 1)$	$7.86 \cdot 10^{-2}$	$4.62 \cdot 10^{-6}$	$9.23 \cdot 10^{-2}$	$5.51 \cdot 10^{-4}$
n_n	$1.01 \cdot 10^{35}$	$3.46 \cdot 10^{33}$	$8.29 \cdot 10^{16}$	$1.23 \cdot 10^{16}$
n_e	$7.18 \cdot 10^{35}$	$3.95 \cdot 10^{35}$	$3.31 \cdot 10^{17}$	$2.20 \cdot 10^{17}$
T_e	$6.74 \cdot 10^0$	$1.11 \cdot 10^{-3}$	$7.24 \cdot 10^{-1}$	$7.66 \cdot 10^{-3}$
$\epsilon_{D\alpha}$	$3.83 \cdot 10^{-1}$		$2.15 \cdot 10^{-1}$	

Table B3: Errors achieved by the optimisation of the parametric maps using the renders and midplane measurements, both in the parametric representation and in the SOLPS representation. MSE is the Mean Square Error and MAE is the Mean Absolute Error.

	Errors on the Modified Experiment iii with $\chi = 0.5$ on Boundary			
	MSE		MAE	
	Whole Map	Midplane	Whole Map	Midplane
$\log_2 (n_{total} + 1)$	$1.55 \cdot 10^{-1}$	$3.88 \cdot 10^{-2}$	$1.62 \cdot 10^{-1}$	$6.05 \cdot 10^{-2}$
χ	$3.00 \cdot 10^{-2}$	$1.18 \cdot 10^{-3}$	$5.65 \cdot 10^{-2}$	$8.13 \cdot 10^{-3}$
$\log_2 (T_e + 1)$	$1.01 \cdot 10^{-1}$	$9.09 \cdot 10^{-6}$	$1.07 \cdot 10^{-1}$	$7.63 \cdot 10^{-4}$
n_n	$1.66 \cdot 10^{35}$	$2.29 \cdot 10^{34}$	$1.19 \cdot 10^{17}$	$3.1 \cdot 10^{16}$
n_e	$7.92 \cdot 10^{35}$	$2.43 \cdot 10^{35}$	$3.53 \cdot 10^{17}$	$1.77 \cdot 10^{17}$
T_e	$6.96 \cdot 10^0$	$1.76 \cdot 10^{-3}$	$7.72 \cdot 10^{-1}$	$9.05 \cdot 10^{-3}$
$\epsilon_{D\alpha}$	$3.54 \cdot 10^{-1}$		$2.04 \cdot 10^{-1}$	