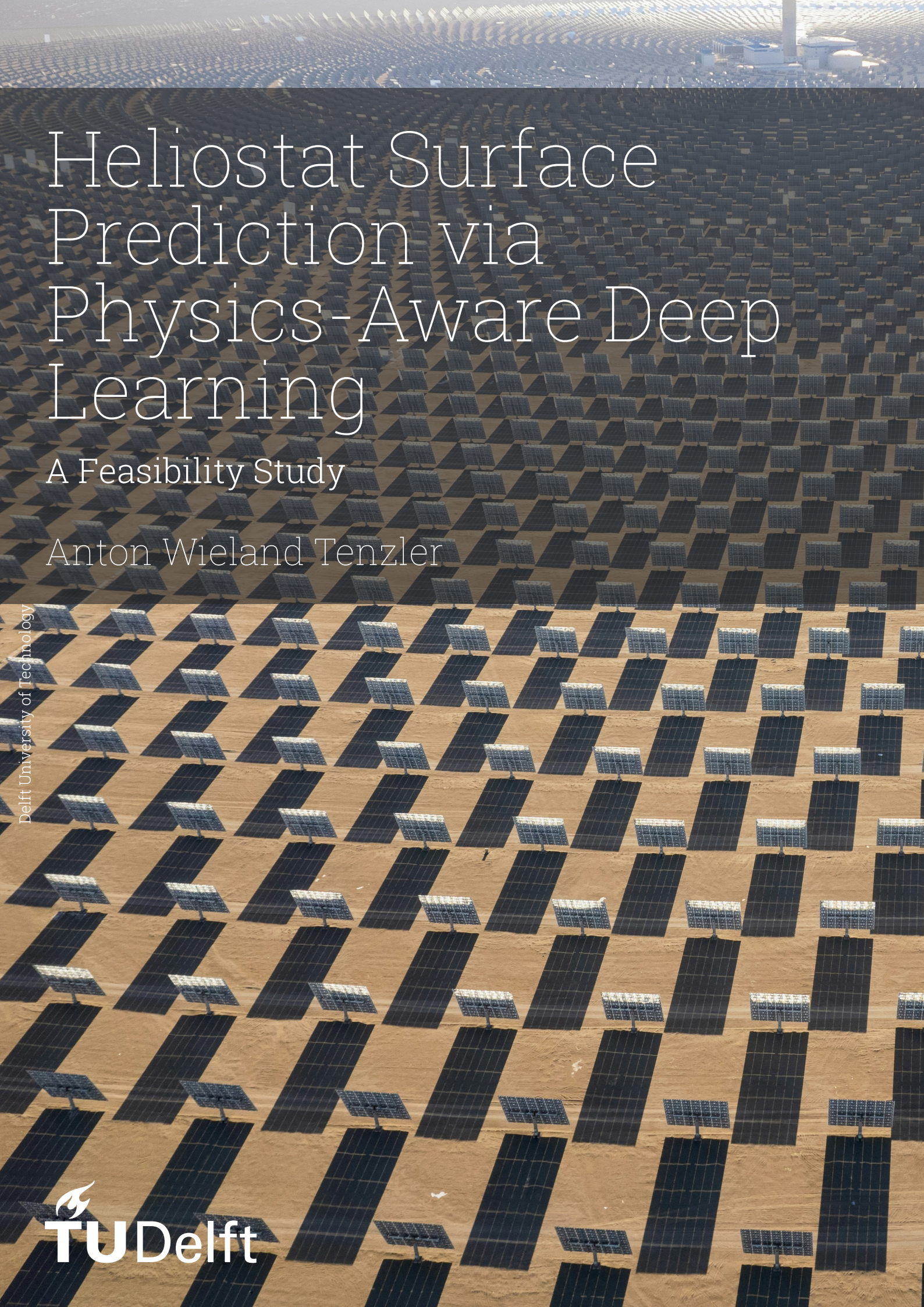


Heliostat Surface Prediction via Physics-Aware Deep Learning

A Feasibility Study

Anton Wieland Tenzler

Delft University of Technology

TU Delft

Heliostat Surface Prediction via Physics-Aware Deep Learning

A Feasibility Study

by

Anton Wieland Tenzler

to obtain the degree of Master of Science
for Sustainable Energy Technology
at the Delft University of Technology,
to be defended publicly on Wednesday August 27, 2025 at 14:00.

Thesis Committee Chair:	Prof. Dr. A. Weeber
Daily Supervisor:	Dr. J. Sun
SET Supervisor:	Dr. M. Vogt
External Supervisors:	Dr. M. Pargmann, Jan Lewen (DLR)
Project Duration:	November, 2024 - August, 2025
Faculty:	EEMCS, Delft

Cover: The cover image shows the Shouhang Dunhuang Phase II CSP plant in China, a 100 MW solar tower facility located in the Gansu province [1].

TU Delft DLR

Preface

This thesis was written in partial fulfillment of the requirements for the Master of Science degree in Sustainable Energy Technology at Delft University of Technology. The work was carried out at the Photovoltaic Materials and Devices (PVMD) group within the Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS), under the supporting supervision of Malte Vogt, with main guidance coming from Jing Sun from the Faculty of Computer Science.

The project was conducted in close collaboration with the German Aerospace Centre (DLR), where I had the pleasure of working under the continuous supervision of Max Pargmann and Jan Lewen. Their expertise and feedback were invaluable in shaping the direction of this research.

I would like to sincerely thank all my supervisors for their consistent support, their readiness to answer questions quickly, and their constructive feedback throughout the course of this work. I am especially grateful for the opportunity to work on a challenging yet highly stimulating thesis topic, which made the past months both demanding and rewarding. I do hope I was able to contribute to this field and that the results of this work may somehow support future research in concentrated solar power technologies.

Anton Wieland Tenzler
Delft, August 2025

Abstract

Accurate characterization of heliostat surface errors is essential for the efficiency of concentrating solar power (CSP) plants, yet direct measurement methods such as deflectometry remain costly and impractical at scale. This thesis investigates a physics-informed deep learning approach to reconstruct heliostat surfaces from flux density images alone—a fundamentally ill-posed problem in which many distinct surfaces can yield similar flux patterns. The proposed framework integrates simulated datasets, augmentation of real surface measurements, and a raytracing-based training loop, with additional regularization strategies to mitigate the ill-posed nature of the inverse problem. The best model achieved a median flux prediction accuracy of 84%, approaching the 92% of supervised benchmarks. For surface reconstruction, training on synthetic datasets with heliostat positions close to the receiver yielded the lowest median Mean Absolute Error (MAE) of 2.4×10^{-4}, compared to 1.4×10^{-4} in the supervised case. While individual surface reconstructions remained limited, the model reproduced some mean structural patterns of the training set, indicating partial learning of underlying geometric behavior. These findings demonstrate both the potential and current limitations of deep learning for heliostat surface reconstruction. With further advances in regularization, dataset design, and real-world validation, the approach may provide a scalable tool for CSP field calibration and optimization in the future.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

Abbreviation	Definition
ARTIST	AI-enhanced Differentiable Ray Tracer for Irradiation Prediction in Solar Tower Digital Twins
CNN	Convolutional Neural Network
CSP	Concentrated Solar Power
DLR	Deutsches Zentrum für Luft- und Raumfahrt (German Aerospace Centre)
DNI	Direct Normal Irradiance
ENU	East-North-Up Coordinate System
GAN	Generative Adversarial Network
ISA	International Standard Atmosphere
LCOE	Levelized Cost of Electricity
LFM	Light Field Microscopy
MAE	Mean Absolute Error
MSE	Mean Squared Error
NN	Neural Network
NURBS	Non-Uniform Rational B-Splines
PAINT	Public Database of Solar Tower Deflectometry Data
PV	Photovoltaic
RMSE	Root Mean Square Error
SSIM	Structural Similarity Index Measure
SVD	Singular Value Decomposition
TES	Thermal Energy Storage

Symbols

Units

Unless otherwise specified, SI units are used throughout. Common derived units include: MWh (megawatt-hour), MW (megawatt), kWh (kilowatt-hour), kW (kilowatt), h (hour), ° (degree), USD (US dollar), m (meter), rad (radian).

CSP Plant Performance

Symbol	Definition	Unit
C_f	Capacity factor of the CSP plant, defined as $C_f = \frac{E_{year}}{P_{rated} \cdot T}$	[%]
E_{year}	Annual energy production	[MWh/year]
P_{rated}	Installed (rated) capacity	[MW]
T	Total hours in one year (typically 8 760)	[h]

Flux Density Prediction Accuracy Metric

Symbol	Definition	Unit
$\mathrm{ACC}_{\mathrm{GT,pred}}$	Relative accuracy metric quantifying deviation between predicted and ground truth flux density images	[-]
ϕ_{GT}	Ground truth flux density	[-]
ϕ_{pred}	Predicted flux density	[-]

Sun Position Calculation

Symbol	Definition	Unit
ϕ	Geographic latitude of observer	[deg]
λ	Geographic longitude of observer	[deg]
ϕ_{rad}	Latitude in radians	[rad]
λ_{rad}	Longitude in radians	[rad]
P	Atmospheric pressure	[bar]
T	Atmospheric temperature	[$^\circ$ C]
UT	Universal Time (decimal hours)	[h]
JD_t	Julian Day offset from J2000.0 epoch	[-]
t	Time variable including Δt correction	[days]
Δt	Time correction (assumed 0)	[s]
θ	Base angular term for heliocentric longitude expansion	[rad]
L	Heliocentric longitude	[rad]
t_2	Scaled time variable ($t/1000$)	[-]
$\Delta\psi$	Nutation correction term	[rad]
ϵ	Earth's obliquity	[rad]
λ_{geo}	Geocentric solar longitude	[rad]
α	Right ascension	[rad]
δ	Declination	[rad]
H	Local hour angle	[rad]
$\Delta\delta$	Parallax correction of declination	[rad]
δ'	Corrected declination	[rad]
$\Delta\alpha$	Parallax correction of right ascension	[rad]
H'	Corrected local hour angle	[rad]
$\theta_{\mathrm{elev,\ no\ refrac}}$	Solar elevation angle without refraction	[rad]
Δ_{refrac}	Atmospheric refraction correction	[rad]
θ_{elev}	Corrected solar elevation angle	[rad]
$\theta^\circ_{\mathrm{elev}}$	Corrected solar elevation angle	[deg]
θ_{azim}	Solar azimuth angle	[rad]
$\theta^\circ_{\mathrm{azim}}$	Solar azimuth angle	[deg]

Data Augmentation

Symbol	Definition	Unit
S_1	Randomly selected heliostat surface	[-]
S_2	Randomly selected heliostat surface	[-]
S_{interp}	Interpolated surface from S_1 and S_2	[-]
α	Random interpolation scalar, $\alpha \in [0, 1]$	[-]

Raytracing Procedure

Symbol	Definition	Unit
$\vec{i}$	Incident ray vector	[-]
$\vec{r}$	Reflected ray vector	[-]
$\vec{n}$	Surface normal vector	[-]
$\vec{n}_{des}$	Desired surface normal vector for heliostat alignment	[-]
$\vec{r}_{des}$	Desired reflection vector from heliostat to aim point	[-]
$\vec{a}$	Aim point coordinates	[-]
$\vec{p}$	Heliostat surface origin (position)	[-]
$\cdot$	Dot product operator	[-]
$\|\cdot\|$	Vector norm (magnitude)	[-]

Regularizations

Symbol	Definition	Unit
Z	z-coordinates of control points for heliostat facets, relative to the ideal reference surface	[m]
$\Delta Z_{i,j}^{(f)}$	Discrete Laplacian at control point (i, j) on facet f (local curvature measure)	[1/m]
n_{bf}	Unit normal vector of facet f in batch b	[-]
θ_{bf}	Tilt angle of facet f in batch b relative to the z-axis	[rad]
Z_{edge}	z-coordinate of an edge control point	[m]
Z_{inner}	z-coordinate of the inner neighbor of an edge control point	[m]
δZ	Edge dip: height difference $Z_{\text{edge}} - Z_{\text{inner}}$	[m]
$L_{\text{z-lim,reg}}$	Regularization loss penalizing violations beyond z_{lim}	[-]
$L_{\text{z-lim}}$	Magnitude of violation beyond z_{lim}	[m]
$\gamma_{\text{z-lim}}$	Weighting factor for z-limit regularization	[-]
z_{lim}	Maximum allowable z-deviation from the reference surface	[m]
L_{L2}	Squared L2 norm of predicted Z values	[m^2]
$\|Z\|_2^2$	Expanded L2 expression, $\sum_{f,i,j} Z_{i,j,f}^2$	[m^2]
$L_{\text{L2,reg}}$	Weighted L2 regularization loss	[-]
γ_{L2}	Weighting factor for L2 regularization	[-]
L_{curve}	Curvature loss penalizing deviations from C_{ref} beyond threshold τ	[1/m^2]
$C_{\text{pred},i}$	Predicted curvature of sample i	[1/m]
C_{ref}	Reference curvature from real-surface analysis	[1/m]
τ	Tolerance threshold for curvature deviation	[1/m]
$\mathcal{I}$	Index set of samples where $\lvert C_{\text{pred},i} - C_{\text{ref}} \rvert > \tau$	[-]
N	Number of samples (or number of stored surfaces, context-dependent)	[-]
$L_{\text{curve,reg}}$	Weighted curvature regularization loss	[-]
γ_{curve}	Weighting factor for curvature regularization	[-]
L_{tilt}	Tilt loss penalizing facet normal deviations from vertical	[rad^2]
$L_{\text{tilt,reg}}$	Weighted tilt regularization loss	[-]
γ_{tilt}	Weighting factor for tilt regularization	[-]
Δ	Edge dip violation measure: $Z_{\text{edge}} - Z_{\text{inner}} + m$	[m]
m	Margin: expected drop in z between inner and outer points	[m]
L_{dip}	Edge dip loss, averaged squared violation across all edges	[m^2]
$E_{i,j}[\cdot]$	Expectation/mean operator along a facet edge	[-]
$\Delta_{\text{top}}, \Delta_{\text{left}}, \Delta_{\text{right}}, \Delta_{\text{bottom}}$	Edge-specific violation terms	[m]
$L_{\text{dip,reg}}$	Weighted edge dip regularization loss	[-]
γ_{dip}	Weighting factor for edge dip regularization	[-]
L_{rep}	Repulsion loss based on cosine similarity to stored surfaces	[-]
$\cos(\theta_i)$	Cosine similarity between current prediction and i-th stored surface	[-]
$\mathbf{z}_{\text{current}}$	Current predicted surface representation	[-]
$\mathbf{z}_{i,\text{past}}$	i-th stored surface in buffer	[-]
$L_{\text{rep,reg}}$	Weighted repulsion regularization loss	[-]
γ_{rep}	Weighting factor for repulsion regularization	[-]

Technical Terms

Term	Definition
Heliostat	Mirror in the solar field reflecting sunlight onto the receiver.
Canting	Process of angling heliostat facets slightly to focus sunlight onto the receiver.
Deflectometry	Optical technique using stripe pattern reflections to measure surface slopes and geometry.
Flux Density Image	Bitmap representation of the reflected solar intensity distribution on a target area.
Ground Truth	Reference or real-world measurement data used to validate predictions.
Neural Network	Machine learning model composed of interconnected layers of neurons.
NURBS	Non-Uniform Rational B-Splines; surface representation using control points, weights, and knot vectors.
Raytracer	Computational tool that simulates the propagation and reflection of rays of light.
Capacity Factor	Ratio of actual energy output to maximum possible energy output over a given period.

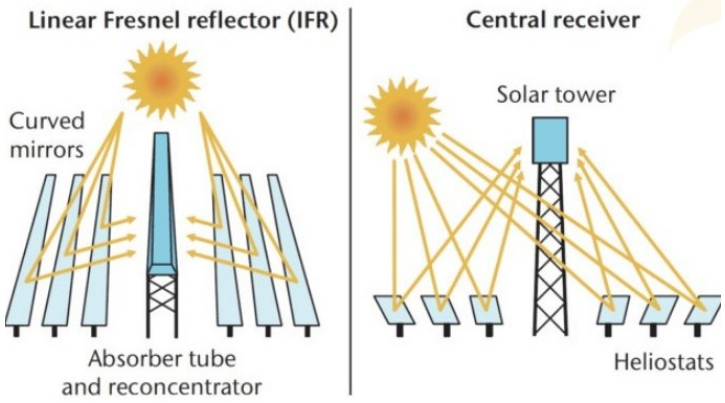
1

Introduction

The global transition toward a low-carbon energy system has become imperative to meet climate targets and deliver energy security and economic growth. Solar energy, as one of the most abundant and rapidly advancing renewable sources, plays a pivotal role in this shift [2]. Within the diverse landscape of solar technologies, Concentrated Solar Power (CSP) offers a distinct approach based on the optical concentration of sunlight and subsequent conversion into thermal energy.

1.1. Concentrated Solar Power

CSP is a method of harnessing renewable solar energy. The configuration of a CSP plant can vary, with four primary technologies currently described in the literature, as illustrated in Figure 1.1. These technologies differ in relative cost, land requirements, operating temperature ranges, and solar concentration ratios [3]. This work focuses exclusively on the solar tower configuration with a central receiver, which still offers significant potential for improvement. In this configuration, the mirrors—known as heliostats—are arranged in varying patterns around the solar tower. The central receiver design provides several advantages over other CSP technologies. Its subsystems, such as heliostats and receivers, can be optimized individually, enabling greater potential for cost reductions and efficiency gains [4]. Further advantages include operating temperatures of up to 1000 $^{\circ}\mathrm{C}$, which improve power conversion efficiency, and high annual capacity factors enabled by integrated thermal storage, as discussed later [4]. Despite these benefits, the parabolic trough configuration remains the most widely deployed CSP technology worldwide [5].



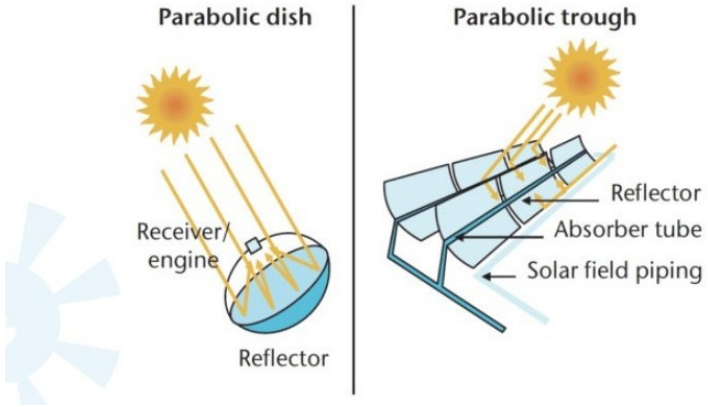


Figure 1.1: Overview of the four main Concentrated Solar Power (CSP) configurations. (a) Linear Fresnel reflector systems use long, slightly curved mirrors to concentrate sunlight onto an elevated absorber tube, often equipped with a secondary concentrator. (b) Central receiver systems (solar towers) employ a field of heliostats that track the Sun and focus radiation onto a receiver at the top of a tower. (c) Parabolic dish systems utilize a dish-shaped mirror to concentrate sunlight onto a receiver at the focal point, typically powering a Stirling engine. (d) Parabolic trough systems employ parabolically curved mirrors that focus sunlight onto a receiver tube along the focal line, with heat transferred by a working fluid circulating through solar field piping. These configurations represent the principal technological pathways for harnessing concentrated solar energy. [6]

In general, a central receiver CSP plant consists of a tower-like structure surrounded by numerous mirrors positioned in its vicinity. These mirrors concentrate solar radiation onto a designated area at the top of the tower, known as the receiver, where the solar energy is absorbed. To maximize absorption and minimize reflection, the receiver is typically coated with a black or dark material [7]. The absorbed energy heats the receiver material, causing its atoms to vibrate. A heat transfer fluid, such as air or molten salt, is circulated past or through the receiver, absorbing the heat primarily via convection [7]. The stored thermal energy in the fluid is then utilized to generate electricity by producing steam for a turbine [6].

It is increasingly common to integrate Thermal Energy Storage Systems (TES) into CSP plants, as illustrated in Figure 1.2. In this configuration, the heat from the transfer fluid is directed into the TES. While various storage technologies exist, sensible heat storage remains the most widely applied, with alternatives such as latent heat or thermochemical storage also under consideration [8]. The integration of TES mitigates the intermittency of energy supply [9].

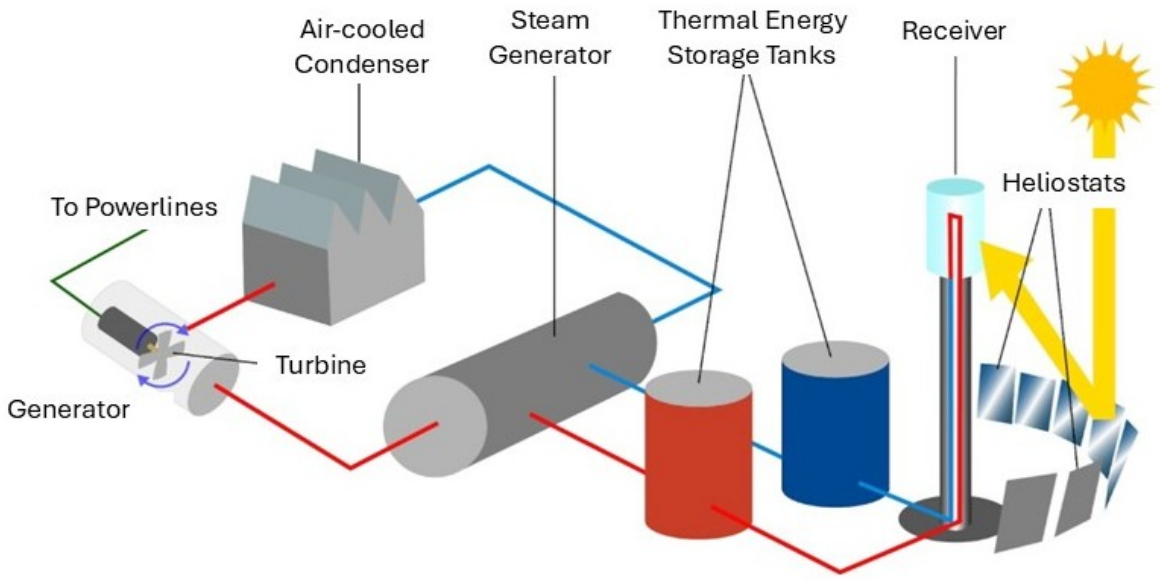


Figure 1.2: Schematic of a central receiver (solar tower) CSP power plant. A field of heliostats concentrates sunlight onto a receiver at the top of the tower, where a heat transfer fluid is heated. The thermal energy can be used immediately to generate steam for a turbine–generator system or stored in thermal energy storage tanks for later use. After passing through the turbine, the working fluid is cooled in an air-cooled condenser and recirculated. This configuration enables both continuous electricity generation and dispatchability through integrated storage. [10]

1.1.1. Evaluation of CSP

Renewable Technologies Economic Comparison

Although research on CSP technologies is ongoing, current market trends indicate that standard photovoltaic (PV) panels offer greater advantages for large-scale, decentralized electricity generation. For instance, the Levelized Cost of Electricity (LCOE) is typically lower for PV systems compared to various CSP technologies. This trend is evident in Table 1.1, which summarizes total installed costs, capacity factors, and LCOE values for different renewable technologies in both 2010 and 2023, including their percentage changes over this period. In comparison with other renewable energy sources—such as onshore and offshore wind, geothermal, hydropower, and bioenergy—the LCOE of CSP plants remained higher in 2023. Another drawback of CSP is its relatively high total installed cost.

Nonetheless, two factors highlight CSP's potential. First, the capacity factor of CSP has experienced the most significant increase and, in 2023, surpassed that of wind and solar PV. Second, the LCOE of CSP has decreased markedly over the years, following a trajectory comparable to the wind technologies. The combination of a high capacity factor and a continuing downward trend in costs suggests that CSP has the potential to regain competitiveness among renewable technologies, particularly when paired with advancements such as TES integration, as will be explored in the following sections.

Table 1.1: Comparison of cost and performance indicators for major renewable energy technologies in 2010 and 2023. Metrics include total installed costs (USD/kW), capacity factors (%), and levelized cost of electricity (LCOE, USD/kWh), along with percentage changes over the period. The data illustrate significant cost reductions in solar PV, onshore wind, and offshore wind, while CSP shows a moderate decline in cost but a substantial improvement in capacity factor. [11]

Technology	Total installed costs (USD/kW)			Capacity factor (%)			LCOE (USD/kWh)		
	2010	2023	% change	2010	2023	% change	2010	2023	% change
Bioenergy	3 010	2 730	-9%	72	72	0%	0.084	0.072	-14%
Geothermal	3 011	4 589	52%	87	82	-6%	0.054	0.071	31%
Hydropower	1 459	2 806	92%	44	53	20%	0.043	0.057	33%
Solar PV	5 310	758	-86%	14	16	14%	0.460	0.044	-90%
CSP	10 453	6 589	-37%	30	55	83%	0.393	0.117	-70%
Onshore wind	2 272	1 160	-49%	27	36	33%	0.111	0.033	-70%
Offshore wind	5 409	2 800	-48%	38	41	8%	0.203	0.075	-63%

For PV systems with battery storage, the LCOE in Germany is projected to range from 0.071 to 0.26 USD/kWh in 2024 [12]. To ensure comparability with Table 1.1, units have been converted from €/kWh to USD/kWh as of July 2, 2025. For large ground-mounted PV plants, which are most comparable to CSP facilities, the LCOE ranges from 0.071 to 0.13 USD/kWh. These values are either equal to or slightly below those of CSP plants. This indicates that CSP has the potential to be competitive in terms of LCOE when storage is included; however, at present, PV systems with batteries generally remain slightly cheaper.

This is because greater investment and research are currently directed toward PV technologies, as their lower manufacturing costs, ease of implementation and operation, and reduced maintenance requirements make them more attractive. In addition, broader expertise and accumulated knowledge exist regarding the installation and application of PV systems [13].

Although PV technologies currently outperform CSP in terms of cost and deployment scale, new CSP projects continue to emerge internationally, underscoring the technology's perceived long-term value. For example, China has 30 CSP plants under construction as of 2025 with a combined capacity of approximately 3 GW [14], while new projects are also progressing in regions such as the Middle East, North Africa, and Spain [15]. These developments highlight sustained international interest in CSP, particularly in locations with high Direct Normal Irradiation (DNI), and indicate that CSP remains part of the broader energy transition despite its current cost challenges.

Geography

CSP technologies have notable geographical limitations. A key disadvantage is that all CSP systems require high levels of DNI and minimal cloud cover, restricting their applicability to specific geographic regions. Figure 1.3 illustrates the global distribution of DNI, highlighting areas with the greatest CSP potential: red and orange regions are highly suitable, yellow regions have moderate suitability, while green regions are not recommended. Consequently, CSP technologies are viable only in limited areas, as regions such as most of Europe, West Africa, East Asia, and the northern parts of both North and South America are not able to exploit CSP effectively.

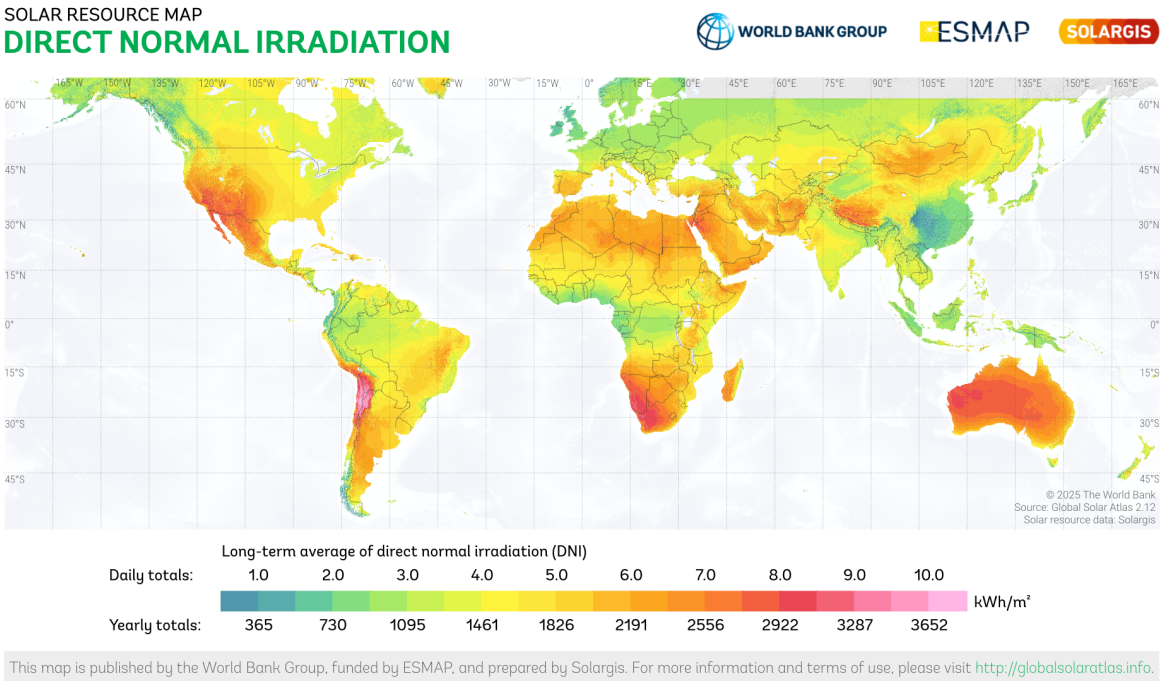


Figure 1.3: Global distribution of DNI based on long-term averages. DNI measures the solar radiation received per unit area by a surface held perpendicular to the Sun's rays and is the key resource indicator for CSP. High-DNI regions, such as North Africa, the Middle East, western China, and Australia, represent prime locations for CSP deployment. [16]

Integrating Thermal Energy Storage

A meaningful cost comparison of solar technologies must also account for intermittency and storage requirements. While CSP remains more expensive than PV in terms of generation alone, the picture changes once storage is included. CSP plants with integrated Thermal Energy Storage (TES) can achieve costs that are comparable to, or lower than, PV systems combined with batteries [13], largely because of the economic and environmental burden of large-scale battery production for PV.

The key advantage of CSP lies in storing energy as heat, typically for several hours depending on the TES configuration. Operational plants worldwide demonstrate storage capacities ranging from 1 to 15 hours [17]. By enabling electricity generation beyond sunlight hours, TES increases annual output and thereby raises the plant's capacity factor. On average, TES integration improves the capacity factor by more than 20% and reduces the LCOE by around 6% [18]. The capacity factor is defined as:

$$C_f = \frac{E_{year}}{P_{rated} \times T} \times 100\% \tag{1.1}$$

where E_{year} is the annual energy production in MWh/year, P_{rated} the installed capacity in MW, and T the total hours in a year (8 760 h).

Beyond power generation, CSP with TES can supply heat directly to industrial processes such as chemical or food production [19], further increasing its value.

Crucially, CSP should not be seen as a competitor to PV but as a complementary technology. While PV provides the cheapest daytime electricity, it remains intermittent and battery-dependent. A combined approach—PV for daytime and CSP for evening and variability—can mitigate these shortcomings [13].

Nevertheless, TES integration is not without challenges. Storage systems add significant cost, increase technical complexity, and remain less mature than PV-battery solutions. This reflects the broader state of CSP, where many baseline technologies are up to two decades old, owing to limited research and development compared with PV [19]. Yet, the performance potential remains evident: large-scale

PV plants in the United States achieve capacity factors between 10% and 36%, averaging 27% [20], whereas CSP plants with TES can exceed 50% [13].

In summary, while CSP with TES still faces technological and economic hurdles, its ability to deliver dispatchable renewable energy and higher long-term capacity factors underscores its role as a critical complement to PV in future energy systems.

1.1.2. Problem Statement

Despite the significant advantages offered by CSP tower systems with central receivers with integrated TES, their overall performance is highly sensitive to the solar flux distribution on the receiver surface. One of the key factors influencing this distribution is the shape and orientation of individual heliostats.

However, determining the actual surface geometry of heliostats in a deployed plant remains a considerable challenge. Current methods for surface measurements are both time-consuming and costly, particularly when applied to large-scale heliostat fields. This limits the potential for dynamic optimization of flux distribution and, consequently, energy yield. This is discussed in detail in Chapter 2.

2

State of the Art

2.1. CSP Plant Jülich, Germany

The Concentrated Solar Power (CSP) plant in Jülich, Germany, operated by the German Aerospace Center (DLR), is a key research facility for the development and validation of solar tower technology. With a nominal capacity of 1.5 MW and a ceramic thermal energy storage system of approximately 1.5 hours, the plant comprises over 2 000 heliostats and provides a versatile platform for experimental studies under realistic operating conditions [21]. A major outcome of this facility is the PAINT database, an open-access, FAIR (Findable, Accessible, Interoperable, Reusable) resource that offers operational data from the Jülich tower. This dataset enables reproducible research, digital-twin development, AI-based calibration, predictive maintenance, and flux simulation studies [22].

Figure 2.1: The Jülich solar tower facility in Germany, operated by the German Aerospace Center (DLR). The facility serves as an experimental platform for CSP technology development and as the source of the PAINT database used in this work.[23]

2.2. Heliostat Surface Measurements in CSP

2.2.1. Importance of Accurate Heliostat Surfaces

Heliostats play a central role in CSP tower systems by reflecting sunlight onto a central receiver, where the resulting flux density distribution directly determines the thermal input and overall energy yield of the plant. Figure 2.2 illustrates the flux density distribution on the receiver of the Jülich tower system. The accuracy of this flux distribution depends not only on the alignment of the heliostats, but also on

the precise geometry of their reflective surfaces. Even small deviations in surface shape can lead to significant distortions in the concentration and direction of reflected sunlight [24].

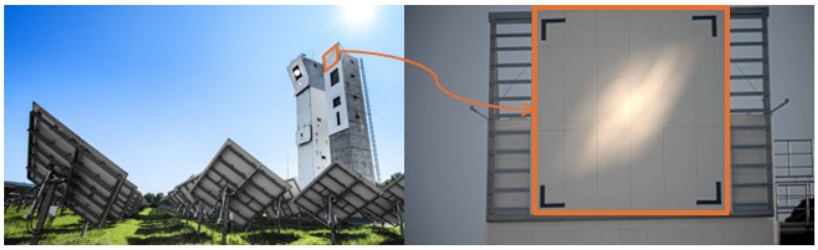


Figure 2.2: Flux density distribution on the test target area of the Jülich solar tower receiver. The figure shows how sunlight reflected by the heliostat field is concentrated onto the receiver surface, producing a distinct flux pattern. [22]

Although heliostats are manufactured within tight tolerance limits, their intended surface geometry cannot always be guaranteed in practice. Furthermore, environmental factors such as wind, temperature fluctuations, and long-term material degradation can introduce surface deformations over time. These imperfections, while often subtle, can substantially impact system performance by altering the optical behavior of the heliostats and reducing the effectiveness of flux concentration on the receiver.

The current state-of-the-art in CSP simulation typically assumes ideal, smooth heliostat surfaces. However, this assumption often results in significant discrepancies between simulated and actual flux distributions. As illustrated in Figure 2.3, even minor surface imperfections can lead to substantial differences in both the shape and intensity of the reflected light. Quantitatively, flux density distributions based on ideal surfaces may achieve only around 67% accuracy when compared to flux distributions based on real measured geometries [24]. These deviations will reduce the system's energy yield. This highlights the need for accurate surface characterization such as deflectometry, which is described in the following section.

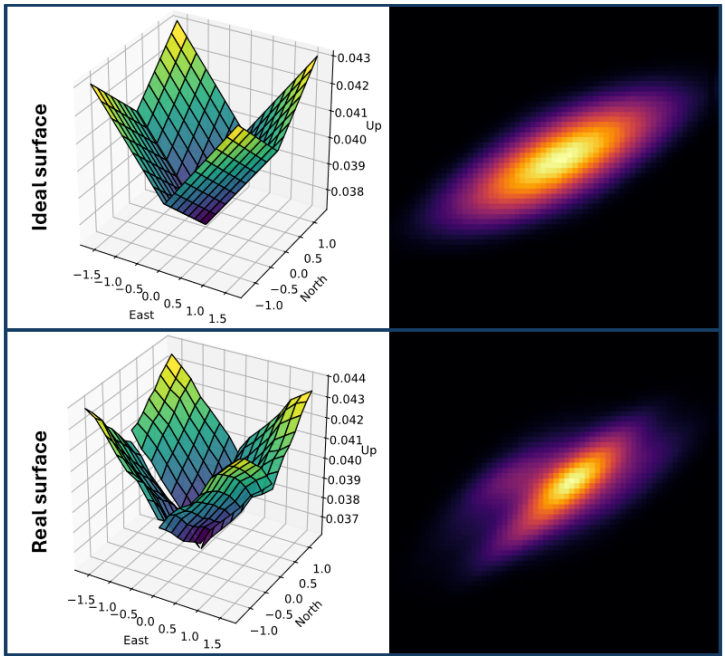


Figure 2.3: Comparison of simulated heliostat surfaces and their corresponding flux density distributions obtained in a ray tracer. The top row shows an idealized, perfectly smooth heliostat surface with the resulting concentrated flux image on the receiver. The bottom row illustrates the real heliostat surface, including geometric imperfections, which lead to a visibly distorted and less uniform flux distribution. This comparison highlights the substantial impact of surface deviations on the accuracy of CSP performance simulations. All axis are in m.

2.2.2. Deflectometry as Surface Measurement Technique

To obtain accurate heliostat surface geometries, deflectometry is currently the most precise and widely used technique. This method involves projecting stripe patterns onto the heliostat surface and recording their reflection using a calibrated camera. By analyzing the deformation of the reflected fringe pattern, local surface slopes can be derived, enabling a detailed reconstruction of the surface [25]. Figure 2.4 shows a typical deflectometry setup and an example of the resulting image. Deflectometry is also widely used in other applications, such as reconstructing and qualifying reflective surfaces in automotive manufacturing [26].

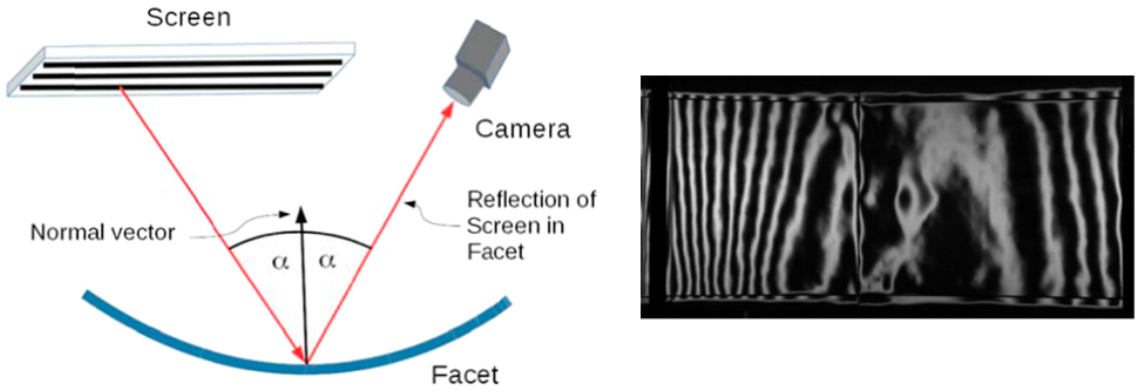


Figure 2.4: Schematic of a deflectometry measurement setup with an example fringe image. A screen projects stripe patterns onto the heliostat surface, which are reflected toward a camera. The deformation of these reflected patterns provides information on local surface slopes, allowing accurate reconstruction of mirror geometries.

Deflectometry has several critical limitations that restrict its practical use in large-scale CSP applications. In the lab, deflectometry has the potential to be efficient, but in deployment under field conditions uncontrollable external issues come into play. Measurements must be conducted at night in the absence of ambient light and are highly sensitive to weather conditions, particularly dew [24]. Furthermore, the process is time-consuming: each measurement requires several minutes per heliostat, and thousands of heliostats must be assessed per plant. At the Jülich facility, for example, measuring all 2 153 heliostats would take more than 100 hours under ideal conditions and assuming 8-hour workdays. In reality, dew makes continuous measurements nearly impossible, with suitable conditions occurring only about once every ten nights. Under these constraints, the total measurement time would increase to approximately 1 076 hours, or roughly 135 workdays.

These logistical and economic constraints make regular deflectometry-based surface reconstruction impractical for large-scale deployment in CSP plants. These limitations have motivated the development of alternative surface characterization methods, such as the deep learning approach suggested by Lewen et al. in Section 2.3.1.

2.3. Deep Learning

Deep learning algorithms have shown exceptional performance across a wide range of domains, including computer vision, speech recognition, and natural language processing [27]. At their core, deep neural networks are capable of approximating complex non-linear functions, with mathematical proofs confirming their universal approximation properties when using two or more hidden layers [27]. A key advantage of deep learning is its ability to process raw input data, such as images, without the need for manual feature extraction. This makes it particularly well suited for regression tasks [28], classification [29], and other image-based analyses [30]. Moreover, deep learning models tend to scale effectively with data volume, continuing to improve in performance as more training data become available—unlike classical models, which often plateau [31].

While training deep neural networks requires large datasets and considerable computational effort, inference using a trained model is typically several orders of magnitude faster than traditional methods [32]. These capabilities position deep learning as a promising tool for addressing current limitations in

CSP surface characterization. In particular, it offers the potential to predict heliostat surface properties from flux density data, reducing dependence on time- and cost-intensive measurement techniques like Deflectometry. The following section introduces a supervised deep learning approach proposed by Lewen et al., which serves as a foundation for the methodology developed in this work.

2.3.1. Supervised Deep Learning for Heliostat Surface Reconstruction

Recent work by Lewen et al. has demonstrated that an inverse deep learning ray tracing approach can be used to accurately predict heliostat surface geometries from flux density images [24]. Their method applies supervised learning, in which a neural network is trained on labeled datasets containing both input features—such as flux density images, Sun positions, and heliostat positions—and corresponding ground truth surface data obtained via deflectometry. Once trained, the network can infer surface geometries for new, unseen data with high accuracy.

However, the reliance on supervised learning introduces a key limitation. Since the training process requires ground truth surface data for each new plant or configuration, deflectometry measurements must still be conducted to generate these labels. As a result, the method cannot fully eliminate the need for costly and time-intensive surface measurements.

Relevant Design Elements from Lewen et al.

While the reliance on labeled training data obtainable only through deflectometry limits the scalability of the supervised approach, several of its core design elements are noteworthy. Lewen et al. employed a specific input structure, a Non-Uniform Rational B-Splines (NURBS) based surface representation, a dedicated neural network architecture, a differentiable ray tracer for flux density image generation and an accuracy metric to quantify the quality of flux simulations. These components are summarized below, along with their advantages and limitations.

Inputs

Lewen et al. employed an input structure consisting of eight Sun positions, eight corresponding flux density images, and the position of the heliostat [24]. An important advantage of this design is that flux density images can be recorded directly at the receiver using a camera or sensor system, making them considerably easier and less expensive to obtain than detailed deflectometry measurements. Furthermore, by including multiple Sun positions, the model benefits from diverse lighting conditions and observation angles. Lewen et al. demonstrated that prediction quality improved with the number of Sun positions and reached a plateau at eight [24].

Sun Positions

To generate Sun positions, Lewen et al. sampled random timestamps throughout the year at the location of Jülich, Germany (latitude 50.92°N, longitude 6.36°E). The uniformly sampled month, day, hour, minute, and second values were then converted into solar elevation and azimuth angles, resulting in realistic distributions of Sun positions. This procedure ensured that the model was trained under representative solar conditions across different times of the year. The detailed calculation can be found in Appendix A.2.

NURBS

In neural network-based surface prediction tasks, the output must be a structured representation of the heliostat surface. A common approach in the current state of the art is to use point cloud representations [33], which capture detailed surface geometry by storing large numbers of surface normal vectors. For example, deflectometry measurements at the Jülich solar tower have produced approximately 80,000 surface normals per facet [24], offering a highly accurate but computationally expensive representation.

While point clouds provide high fidelity, they pose practical challenges for machine learning applications due to their large parameter count and memory requirements. Incorporating such detailed output representations into a neural network would significantly increase computational load during training and inference, making them unsuitable for efficient optimization workflows.

To reduce complexity, many geometric modeling applications—including prior work in CSP surface modeling—employ NURBS to represent complex surfaces with significantly fewer parameters [34, 35]. A NURBS surface is defined by a structured grid of control points, along with knot vectors, polynomial

degrees, and associated weights, enabling efficient and flexible surface representation at substantially lower computational cost.

In the context of heliostat surface modeling, a 4-facet heliostat can be approximated using a grid of 8×8 control points per facet, resulting in 256 total control points. Compared to the original 80 000 surface normals per facet, this corresponds to a reduction of over 99.97% in output dimensionality [24], making the approach highly compatible with deep learning models. Figures 2.5a to 2.5c illustrate the comparison between point cloud and NURBS representations. As seen in the point cloud, the surface is captured with high precision but at the cost of requiring a very large number of parameters. Figure 2.5b shows the 8×8 control point representation for each facet. Although this representation provides less detail than the point cloud, it still preserves the overall shape of the surface and can approximate the real surface shown in Figure 2.5c with sufficient accuracy.

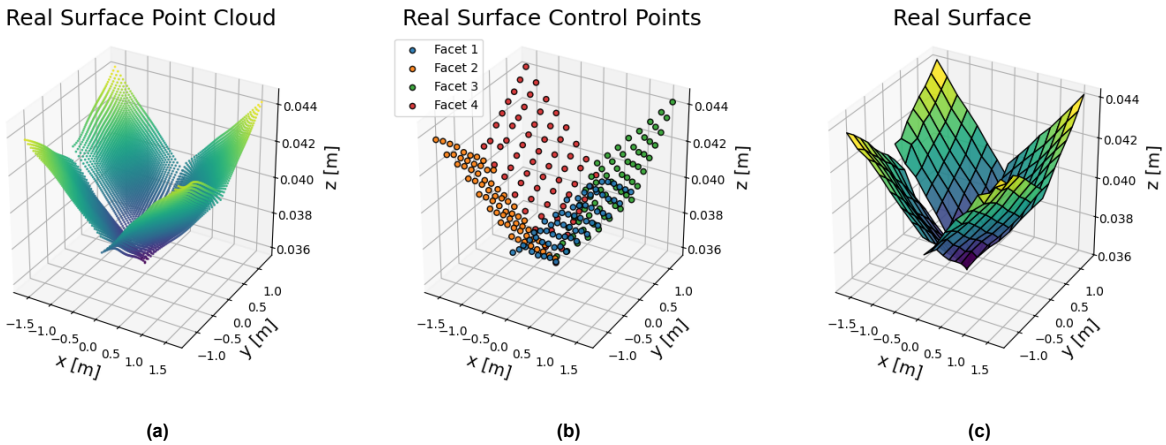


Figure 2.5: Representation of a heliostat surface using different geometric formats. (a) A point cloud obtained from measurements, containing tens of thousands of surface normals. (b) A reduced set of control points used for a NURBS-based surface representation. (c) The reconstructed surface after applying the NURBS model, which approximates the measured geometry with far fewer parameters while retaining overall shape and accuracy.

Neural Network Architecture

Lewen et al. [24] developed a neural network architecture with over four million trainable parameters, shown in Figure 2.6. The model follows an encoder–decoder structure designed to process heterogeneous input types and produce structured surface predictions.

The encoder processes two types of input data: eight flux density images with a resolution of 64×64 pixels, and scalar 3D coordinates for both Sun and heliostat positions. The image inputs are passed through a series of convolutional layers, while the scalar inputs are integrated into the network using a technique known as weight demodulation, as described in [36]. The encoder compresses this input into a latent space W^+, which contains the essential information required to reconstruct the heliostat surface.

By reducing input dimensionality, the encoder enables efficient training while preserving relevant image features such as brightness distributions and energy patterns. The decoder reconstructs the surface geometry from the latent representation and is based on the generator component of the StyleGAN2 architecture. This design allows the network to output NURBS control points directly, combining efficiency with flexibility. Further details on the architecture and StyleGAN2 framework are given in [24, 36].

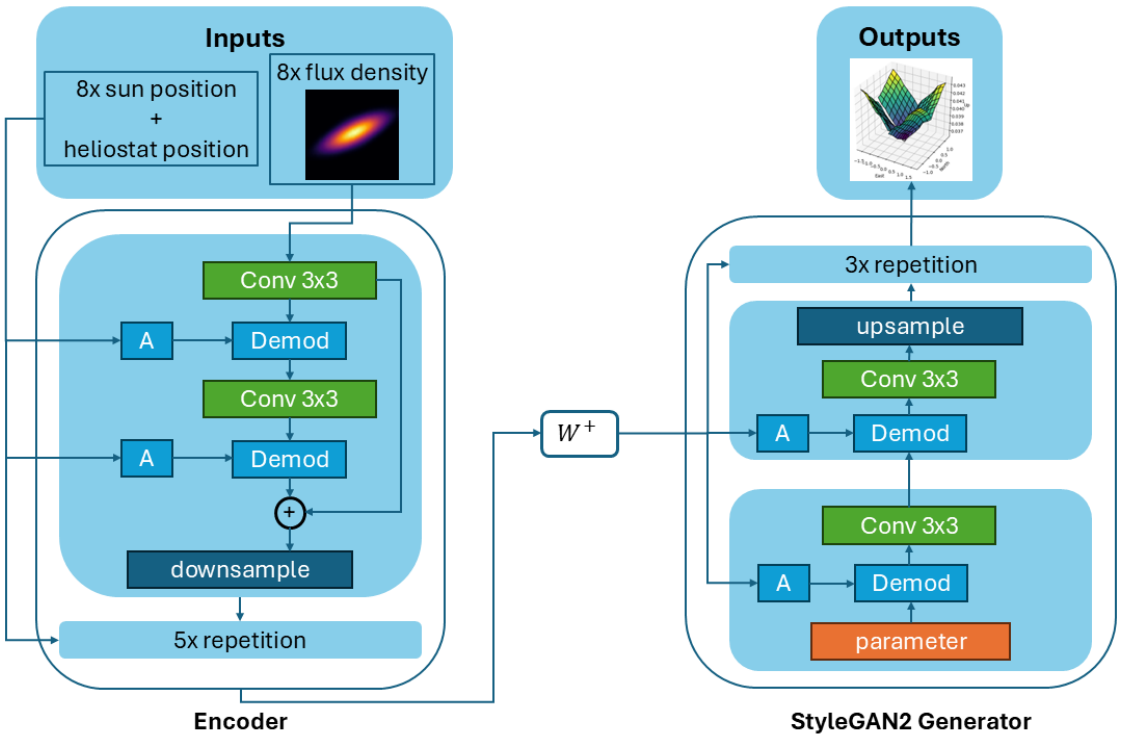


Figure 2.6: Encoder–decoder network architecture for heliostat surface prediction developed by Lewen et al. [24]. The encoder processes heterogeneous inputs, including eight flux density images and scalar Sun and heliostat positions, and compresses them into a latent representation W^+. The StyleGAN2-based decoder reconstructs the heliostat surface from this latent space, outputting structured NURBS control points.

Raytracing

The ray tracer employed by Lewen et al. to generate flux density distributions is part of the *AI-enhanced Differentiable Ray Tracer for Irradiation Prediction in Solar Tower Digital Twins* (ARTIST) project [37]. Ray tracing simulates the physical behavior of sunlight by modeling the paths of individual rays. For a given Sun position, rays are traced as they travel in straight lines to the heliostat, where they are reflected according to the angle of incidence and the heliostat's orientation, before being directed toward a predefined target area. Owing to its differentiability, the ARTIST ray tracer enables gradients to be propagated through the simulation, making it particularly well suited for integration with deep learning approaches such as the one proposed by Lewen et al.

Raytracing Scene

Since the rays are traced from the source (Sun) to the sensor (target area), this corresponds to forward ray tracing. All the variables describing the overall CSP system are saved in the raytracing scene. It consists of a central receiver, a set of heliostats positioned at predefined locations, a fixed target area where the reflected rays are directed, and a Sun position vector. The ray tracer simulates the propagation of rays from the Sun direction toward the heliostat field, where reflections are computed based on the surface geometry and orientation of each heliostat. The resulting flux density image is then calculated from the distribution of these reflected rays on the target area. Further details of the procedure are documented in Appendix A.4.

Raytracing Parallelization

The raytracer is highly parallelized, meaning that it can generate flux densities for several surfaces with the same Sun position at the same time. This reduces runtime significantly for simulations with several heliostats. A limitation is however that it cannot run in parallel to generate flux densities for several different Sun positions with the same surface.

Flux Density Prediction Accuracy Metric
Lewen et al. introduced a relative accuracy metric to evaluate the quality of simulated flux density images. This metric is defined as

$$\mathrm{ACC}_{\mathrm{GT,pred}} = \frac{\sum |\phi_{\mathrm{GT}} - \phi_{\mathrm{pred}}|}{\sum |\phi_{\mathrm{GT}}|}, \tag{2.1}$$

where ϕ_{GT} denotes the ground truth flux image and ϕ_{pred} the predicted flux image. The denominator corresponds to the total energy in the ground truth image.

To ensure consistency across samples, Lewen et al. normalize all flux density images such that their total energy equals one:

$$\sum |\phi_{\mathrm{GT}}| = 1. \tag{2.2}$$

This metric directly quantifies the relative deviation of predicted flux from the ground truth.

Summary
In summary, Lewen et al. demonstrated that heliostat surface reconstruction can be approached effectively with a combination of carefully designed inputs, a compact NURBS surface representation, a dedicated encoder–decoder network, and a differentiable ray tracer. An important advantage of their approach is the use of flux density images as inputs, since these can be obtained at low cost via camera directly at the receiver. However, the scalability of the method remains limited by its reliance on deflectometry data as ground-truth supervision during training. This highlights a key open problem: whether surface prediction can be achieved based solely on flux measurements, without the need for deflectometry.

Yet, relying solely on flux density for supervision introduces a fundamental difficulty: the inverse mapping from flux density distributions to heliostat surface geometries is strongly ill-posed. Multiple different surface shapes can generate nearly identical flux patterns, making it impossible to uniquely recover the underlying geometry without additional ground-truth information.

2.3.2. Ill-Posed Inverse Problems

Physics-Driven Approaches in Geophysics
Similar challenges are found in geophysics, where inverse problems are pervasive. Sun et al. address the removal of free-surface multiples in seismic data, where the recorded wavefield consists of both useful primaries and unwanted multiples. Since clean primary data are not available for training, they propose a physics-driven self-supervised Convolutional Neural Network (CNN) that predicts the primaries while treating the multiples as the residual between the prediction and the full wavefield. Their network is trained by enforcing that the predicted primaries plus the residual multiples must reconstruct the original recorded data. This formulation embeds the governing physical relation directly into the loss function, enabling effective multiple elimination without requiring labeled training data [38].

In another study, Sun et al. tackle the ill-posed problem of downward continuation in potential field geophysics, where measurements taken at the Earth's surface must be projected downward to estimate the field closer to the source. Because small measurement errors can destabilize this process, Sun et al. introduce a physics-trained neural network that incorporates the known forward operator, upward continuation, directly into the training loop. Any predicted downward-continued field must, when continued upward, exactly reproduce the observed surface data. This constraint allows the network to learn stable inverse solutions without paired ground-truth labels [39].

Physics-Driven Approaches in Other Fields
Beyond geophysics, similar physics-driven learning strategies have been applied successfully in other domains facing ill-posed inverse problems.

In acoustics, Luan et al. proposed a physics-informed neural network for Near-Field Acoustic Holography, where the task is to reconstruct vibrating surface velocity fields from measured sound pressures in

the air. This inverse problem is highly unstable, as small measurement errors can lead to large reconstruction errors. Their approach integrates the Kirchhoff–Helmholtz integral—the governing physical relation between surface vibrations and acoustic fields—directly into the loss function. Instead of training on paired “surface–sound” datasets, the network predicts surface vibrations whose forward-propagated sound field must match the recorded measurements. This physics-driven supervision enables accurate reconstructions without requiring labeled training data [40].

A related idea has also been applied in biomedical imaging. Lu et al. introduced SeReNet for light-field microscopy (LFM), which captures four-dimensional optical information but requires computationally demanding reconstruction to obtain three-dimensional images. Instead of relying on large labeled datasets of LFM measurements and corresponding ground-truth 3D structures, SeReNet enforces that its predicted 3D reconstructions, when passed through the known LFM imaging model, reproduce the originally measured raw light-field data. In this way, the physics of image formation itself provides the supervisory signal. The result is rapid and accurate in obtaining 3D reconstructions without the need for costly experimental training pairs [41].

These examples reinforce the central insight that embedding physics-based forward models into the training process can overcome the lack of labeled data, yielding stable solutions to inverse problems in diverse scientific domains. This paradigm is directly transferable to heliostat surface reconstruction, where the governing physics of light reflection and flux propagation can serve as implicit supervision in the absence of ground-truth deflectometry.

Physics-Driven Framework for Ill-Posed Inverse Problems

The previously listed works demonstrate that ill-posed inverse problems can be addressed using a common physics-driven framework, as illustrated in Figure 2.7. In this scheme, measured data are fed into a neural network that predicts the hidden physical quantity of interest. A differentiable forward model, representing the governing physics of the system, then maps this prediction back into the measurement space. The discrepancy between simulated and observed measurements defines the training loss, which is propagated back through the network. This approach has been applied successfully across diverse domains: seismic imaging, where primaries are separated from multiples using the wave equation [38]; potential field geophysics, where downward continuation is stabilized by embedding upward continuation operators [39]; acoustics, where vibrating surface velocities are reconstructed from sound pressures via the Helmholtz integral [40]; and microscopy, where 3D structures are inferred from raw light-field measurements using the imaging forward model [41]. The repeated success of this architecture across fields highlights its role as a generalizable framework for ill-posed inverse problems, and it forms the conceptual foundation for the approach developed in this thesis.

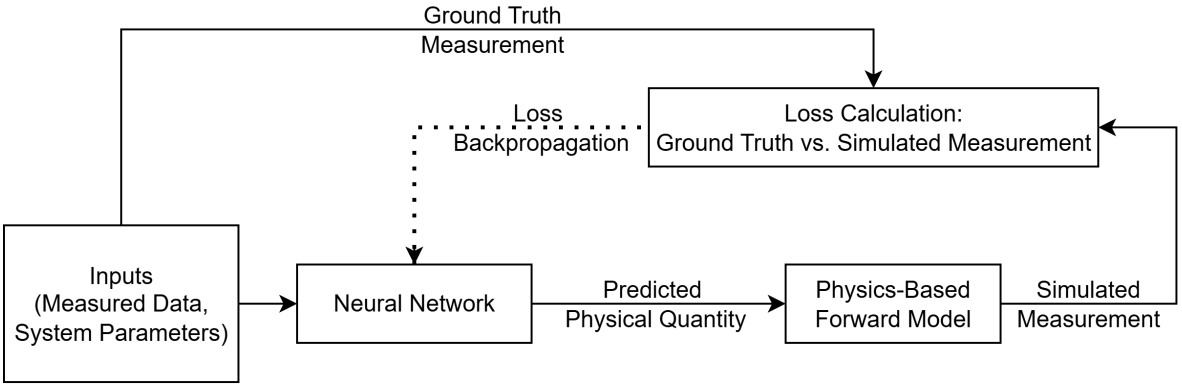


Figure 2.7: General physics-driven framework for solving ill-posed inverse problems. Measured data are provided to a neural network, which predicts the hidden physical quantity of interest. A physics-based forward model then maps the prediction back into the measurement space, where it is compared to the observed data. The resulting discrepancy defines the loss function for training, enabling learning without explicit ground-truth labels.

3

Research Aim

While CSP offers advantages such as integrated thermal storage, it has so far lagged behind PV systems in terms of global deployment and cost efficiency. Nevertheless, large-scale CSP plants continue to be employed, and as the technology continues to mature, further performance gains are expected to arise from computational optimization. One key opportunity lies in flux density optimization at the receiver of tower-based systems, which directly impacts thermal efficiency and energy yield.

In these systems, heliostats reflect sunlight onto a central receiver. The resulting flux density distribution on the receiver is highly sensitive to the surface geometry of each heliostat. Accurate knowledge of these surfaces is therefore essential for predicting and optimizing performance. Heliostat surfaces are often modeled as perfectly smooth and idealized, a simplification that introduces significant errors.

To address this, deflectometry is currently used to measure heliostat surfaces. While precise, the method is logistically and economically impractical at scale. The measurement process is complex and requires very specific circumstances, limiting the feasibility of continuous monitoring or deployment in large-scale CSP systems.

Deep learning offers a potential alternative. Lewen et al. demonstrated that a supervised neural network can infer heliostat surfaces from flux density images and system parameters—but only when trained on ground-truth surface data obtained through deflectometry [24]. As a result, their method still depends on deflectometry, and does not eliminate the associated limitations.

This opens the question whether deflectometry could be omitted entirely, and surface prediction achieved using only flux density images. Since these measurements can be collected cheaply and continuously at the receiver, such an approach would be far more practical and scalable. However, relying solely on flux data introduces a fundamental difficulty: the inverse mapping from flux density to surface geometry is strongly ill-posed. Multiple different surfaces can produce nearly identical flux patterns, making it impossible to uniquely recover the underlying geometry without additional ground-truth constraints.

3.1. Framework

The goal of this work is to eliminate the need for deflectometry data in heliostat surface prediction for CSP systems, while maintaining accuracy comparable to the supervised approach of Lewen et al. To this end, the proposed method builds on their architecture but removes the requirement for labeled surface data. Instead, it predicts heliostat surfaces using only data readily available in the system—namely, Sun positions, heliostat positions, and flux density images. As illustrated in Figure 3.1, this predicted surface is passed through a physics-based ray tracer, which simulates the corresponding flux distribution; this simulated image is then compared to the input image, and the difference is used as the training signal.

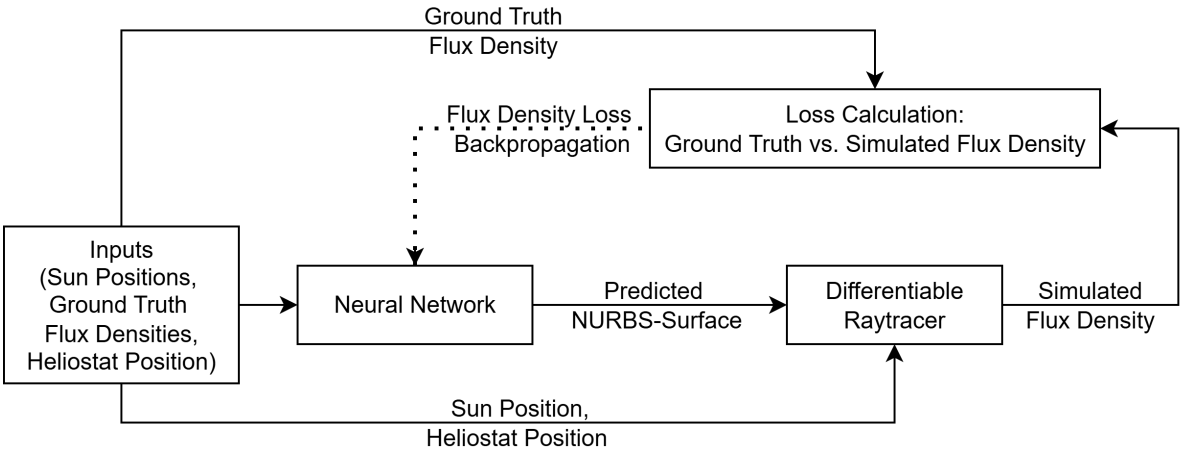


Figure 3.1: Schematic of the physics-aware training loop. The neural network predicts heliostat surface geometry from Sun positions, heliostat position, and flux densities. The predicted surface is passed to a differentiable ray tracer to generate a simulated flux density image. A loss function compares the simulated and ground truth flux density images, and the error is backpropagated through the network. This iterative loop continues until convergence.

This setup follows the general physics-driven framework for ill-posed inverse problems, as demonstrated in domains such as seismic imaging, potential field geophysics, acoustics, and microscopy. By embedding a differentiable forward model into the training loop, the network is constrained to produce predictions that remain consistent with the governing physics. However, in the heliostat setting the inverse mapping from flux density to surface geometry is particularly ill-posed, since multiple distinct surface configurations can give rise to nearly identical flux distributions. This makes the problem more ambiguous than in many related applications, motivating additional regularization strategies to be developed and applied in this work.

The next chapter details the methodology developed to realize this framework, including the dataset construction, training process, and incorporation of further physics-based constraints to address the ill-posedness.

4

Methodology

This chapter presents the methodology used to implement the proposed physics-aware deep learning framework. The chapter is divided into two main parts. The first part outlines the conceptual design of the approach, including the definition of inputs, the neural network architecture, the integration of the differentiable ray tracer, and the evaluation metrics and constraints applied to guide learning. The second part focuses on the practical implementation of this methodology in Python, illustrated through flow diagrams of the primary scripts and the data preprocessing pipeline. Together, these two parts provide both the theoretical and technical foundations for the experiments presented in Chapter 5.

4.1. Surface Prediction via Physics-Aware Deep Learning

The overall training procedure developed in this work is illustrated schematically in Figure 3.1. The following sections describe each component of this pipeline in sequence, beginning with the input stage.

4.1.1. Inputs

Each training sample corresponds to a single heliostat and its reflection behavior under a specific solar configuration. In the following, the term *training sample* refers to one complete input to the network, consisting of a heliostat position, eight Sun positions, and the corresponding eight flux density images.

Sun Positions

In the proposed training pipeline, Sun positions constitute a key component of the neural network inputs, as they define the incident direction of solar radiation for each simulated heliostat configuration. To ensure consistency across the dataset and compatibility with the subsequent physics-based simulations, each Sun position is represented as a unit vector in the East–North–Up (ENU) coordinate system, with the origin defined at the base of the solar tower.

An example distribution of the randomly selected Sun positions is shown in Figure 4.1. Each position is defined by a pair of azimuth (α) and elevation (ϵ) angles, computed as described in Appendix A.2. The azimuth and elevation pairs are converted into a unit vector according to Equation 4.1:

$$\vec{v} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos(\epsilon) \cdot \cos(\alpha) \\ \cos(\epsilon) \cdot \sin(\alpha) \\ \sin(\epsilon) \end{bmatrix} \tag{4.1}$$

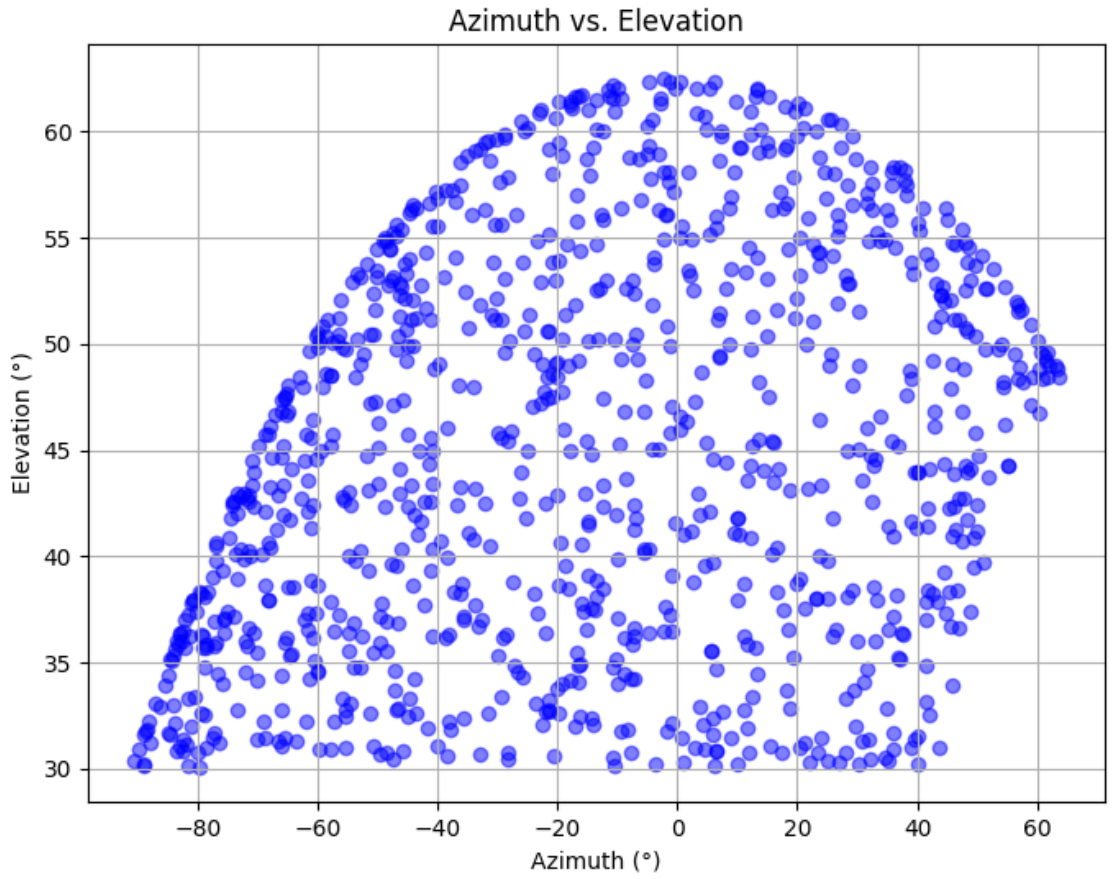


Figure 4.1: Distribution of simulated Sun positions in azimuth–elevation space, representing randomly sampled conditions over the course of a year at the Jülich tower site. The sampling ensures coverage of the solar path relevant for heliostat field operation and provides diverse input conditions for neural network training.

While the full annual range of Sun positions is theoretically available (Section 2.3.1), certain positions pose a challenge for accurate simulation in this work. In particular, Sun positions with an elevation below $30°$ or with a north component of the converted unit vector exceeding 0.6 result in ray tracing errors for heliostats located in the first row of the grid, causing the target area on the tower to be partially or completely missed. Such positions do not provide meaningful training data and are therefore excluded from the dataset, as can be seen in Figure 4.1.

After filtering, the remaining valid Sun positions are used to generate the corresponding flux density images, which, together with heliostat position data, form the complete training sample set for the neural network.

Flux Density Images

In the proposed training framework, each heliostat–Sun position pair is represented by a corresponding flux density image. For every heliostat, flux density images are generated for eight distinct Sun positions using a differentiable ray tracer, which is available as open-source code via GitHub [37] and described in detail in Section 2.3.1. Each image has a resolution of 64×64 pixels and is stored in bitmap format, where pixel intensities are represented numerically and the complete image is encoded as a tensor. An example set of eight flux density images for a single heliostat is shown in Figure 4.2, illustrating the variation in flux distribution with changing Sun positions.

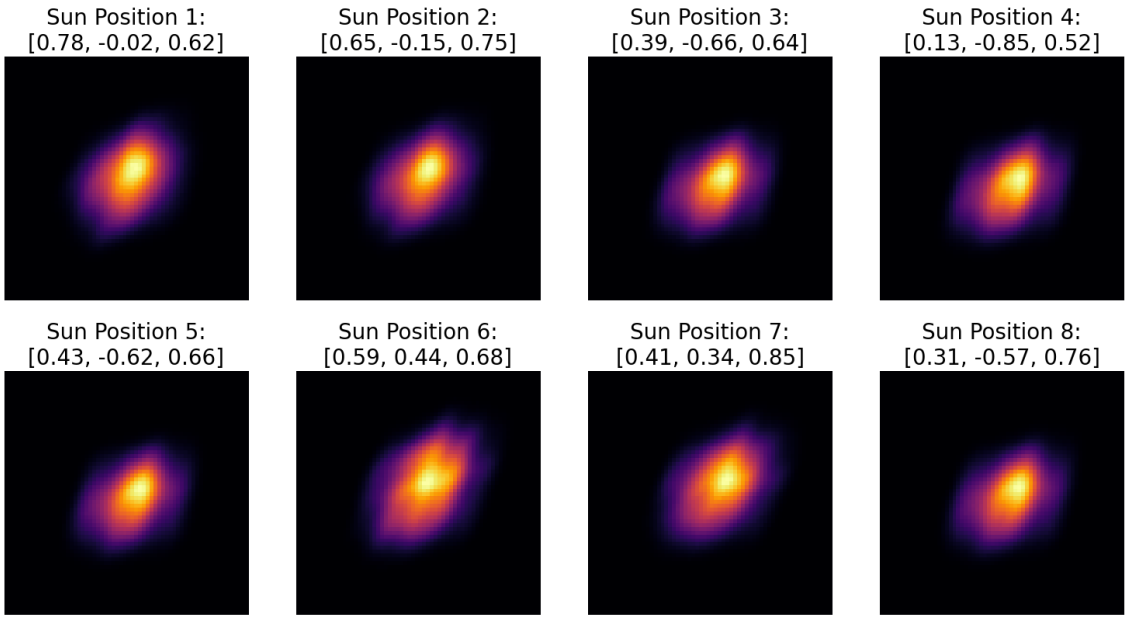


Figure 4.2: Example flux density images for a single heliostat under eight distinct Sun positions. Each image corresponds to one ENU unit vector listed above the panel. The variation in the flux distributions illustrates how changes in solar position affect the reflected energy pattern, forming the basis for the image-based inputs used in the neural network training.

Since the surface prediction task is inherently ill-posed, incorporating multiple Sun positions per heliostat helps constrain the solution space. Using several flux density images provides two main benefits. First, it increases the amount of information available to the network during training. Second, it captures the variation in flux distributions caused by changes in illumination conditions for the same surface.

Based on prior findings by Lewen et al. [24], the number of Sun positions was set to eight. This value offers a favorable trade-off between predictive accuracy and computational cost: while additional Sun positions could in principle provide more information, their benefit saturates around eight and runtime increases substantially. Later experiments (Appendix A.3) confirm that eight Sun positions improve both flux image accuracy and surface prediction compared to single-Sun setups.

In addition to the flux density images, spatial information is required to define the location of each heliostat within the field. This positional data is described in the following section.

Heliostat Positions

Each training sample includes the position of the corresponding heliostat in ENU coordinates. The actual heliostat positions from the Jülich power plant field are used for this purpose. Figure 4.3 shows the layout of the heliostat grid employed during training. The inclusion of positional information is essential, as the geometric relationship between the heliostat, the Sun, and the receiver has a direct impact on the resulting flux density distribution.

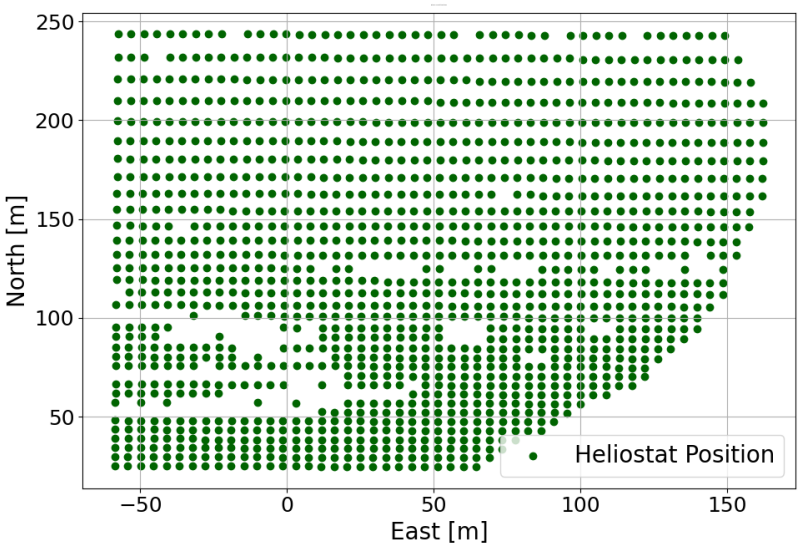


Figure 4.3: Layout of the heliostat field at the Jülich solar tower, shown in East–North–Up (ENU) coordinates. Each green marker corresponds to the position of an individual heliostat used in the dataset.

In addition to spatial position, the structural configuration and orientation of each heliostat influence the flux density distribution. Each heliostat in the Jülich CSP plant is rectangular and composed of four flat rectangular facets, as shown in Figure 4.4. Since the facets are flat rather than curved, they must be slightly angled towards each other to focus sunlight onto the receiver — a process referred to as canting. The canting angles depend on the heliostat's location within the field and are represented by canting vectors, which define the direction of reflected rays toward the target area on the tower.

Figure 4.4: Example of a real heliostat at the Jülich CSP plant. Each heliostat consists of four flat rectangular facets that are slightly canted toward one another to concentrate sunlight onto the receiver. [23]

Certain heliostats located in the central area of the field are aimed at alternative targets for testing purposes and therefore exhibit different canting values in the available deflectometry data. To maintain a consistent dataset and reduce unnecessary variability in the training process, these 130 heliostats are excluded, leaving 1 474 positions for training and data augmentation. This also explains the gaps visible in the field in Figure 4.3.

While this work primarily uses the real heliostat positions from the Jülich plant, alternative synthetic positions are also explored to assess their impact on prediction performance. These variations are presented and discussed in detail in Section 4.1.9.

The next step in preparing the training sample concerns the quantity and diversity of available data,

which directly influence the neural network's capacity to generalize. Therefore, the subsequent section examines the dataset size and augmentation strategies applied in this work.

Augmentation of Input Data

In addition to the characteristics of the input features, the size of the dataset plays a decisive role in the effective training of a neural network. For the Jülich power plant, deflectometry measurements are available for only 423 heliostats. This dataset is publicly accessible through the PAINT database. As described above, eight Sun positions are considered per heliostat. Using the ray tracing procedure to generate the corresponding flux density images from these 423 measured surfaces yields 3 384 images derived from real heliostat data.

This quantity is far below the dataset size typically required for training deep learning models. For reference, Lewen et al. [24] employed approximately 160 000 surfaces for their experiments. Moreover, not all available real surfaces can be allocated to training, as a portion must be set aside for testing to ensure comparability with previous work. Specifically, 56 heliostat surfaces (13.2% of the real dataset) are reserved for testing, matching the selection of heliostats used in [24]. The remaining 367 real surfaces are available for training, which remains insufficient for robust model optimization without augmentation.

To address this limitation, a two-stage augmentation procedure is applied to generate a synthetic dataset of 50 000 surfaces while preserving physical plausibility:

1. **Rotation:** All existing surfaces are rotated by 180°, effectively doubling the number of surfaces.
2. **Interpolation:** Pairs of real surfaces are randomly selected, and an interpolated surface is generated according to:

$$S_{\text{interp}} = \alpha \cdot S_1 + (1 - \alpha) \cdot S_2, \quad \alpha \in [0, 1] \tag{4.2}$$

Here, S_1 and S_2 are the randomly selected surfaces, S_{interp} is the resulting interpolated surface, and α is a random scalar drawn from the interval $[0, 1]$.

All augmented surfaces share identical x and y coordinates, while only the z-coordinates—representing deviations from the ideal flat surface—are varied.

Once the augmented surfaces are generated, they are integrated into the grid of real heliostat positions to create realistic but simulated training samples. For each sample, a random heliostat position is selected from the grid of real positions shown in Figure 4.3. The augmented surface is then added to the ideal heliostat corresponding to this position in the grid, resulting in a realistic but simulated surface. The canting vector described in Section 4.1.1 is already incorporated into the ideal heliostat model for each position, meaning that every heliostat position has its own unique ideal surface. Figure 4.5 illustrates an example of this addition process.

To maintain a strong influence of authentic measurements, the final training dataset is composed of 20% real surfaces and 80% augmented surfaces. This results in repeated use of real surfaces, but each is combined with different positions within the field, increasing diversity while preserving physical realism. The purpose of this approach is to assign greater weight to the real surfaces, ensuring that the neural network is trained with a substantial fraction of authentic data. The goal is to maintain the training dataset as realistic as possible while significantly increasing its size.

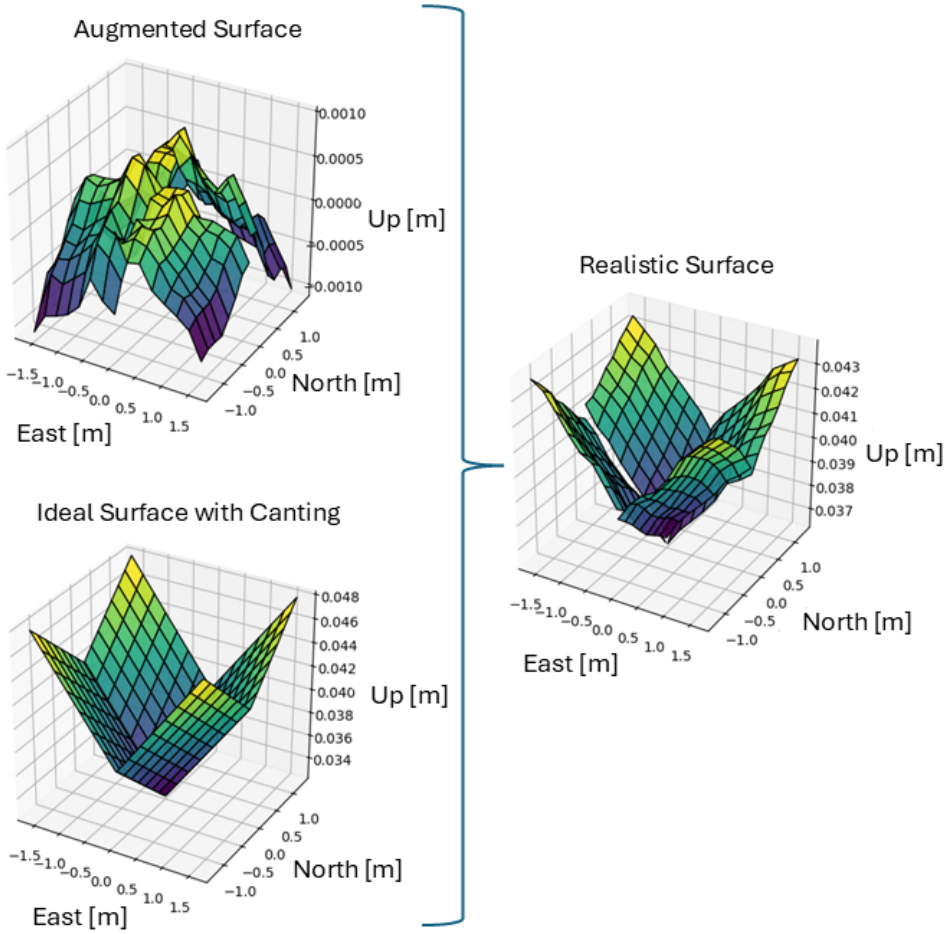


Figure 4.5: Illustration of generating a simulated realistic heliostat surface by combining an augmented surface with the ideal heliostat geometry. The ideal surface, including canting of the four facets, is modified by adding small-scale deviations represented by the augmented surface. The resulting composite surface preserves the x–y geometry of the ideal heliostat while altering the z-coordinates, yielding a realistic approximation of measured mirror imperfections.

4.1.2. Neural Network

In brief, the neural network processes both image-based flux density inputs and scalar Sun and heliostat positional data to reconstruct surface geometry. Its architecture follows the encoder–decoder design of Lewen et al. [24] (Section 2.3.1).

Hyperparameter Tuning

While the network architecture is fixed to ensure comparability with prior work, its performance depends strongly on the choice of training hyperparameters. These parameters govern how the model learns from data and can significantly influence both convergence stability and final reconstruction accuracy. To identify an effective configuration, hyperparameters are optimized in a two-stage process.

In the first stage, a grid search is conducted using a reduced dataset to limit runtime. This step provides an initial set of working parameters, summarized in Table A.3. This stage is going to be referred to as the standard hyperparameter configuration.

In the second stage, a Bayesian hyperparameter optimization explores a broader parameter space—including learning rate, weight decay, error criterion, and learning rate scheduler settings (see Table A.2)—to identify configurations that match or even exceed grid search performance. Unlike grid or random search, Bayesian hyperparameter optimization uses a probabilistic model of the objective

function—typically a Gaussian Process—to model the relationship between hyperparameters and validation performance. This surrogate model predicts both expected performance and uncertainty, enabling efficient decision-making via acquisition functions that balance exploration and exploitation. This approach has been shown to substantially reduce the number of expensive training runs required to discover effective configurations and, in some cases, to surpass expert-level tuning [42]. This approach enables more efficient and targeted hyperparameter tuning, crucial when each evaluation requires full training and validation of a complex deep model.

For both optimization methods, 200 training samples were used. This reduced dataset size was chosen to allow for a faster comparison across a wider range of hyperparameters. The best-performing configurations identified in this step are later validated on larger datasets. The results of both approaches are listed in Table A.3, and both configurations are carried forward for further testing.

4.1.3. Outputs

With the neural network architecture and its hyperparameters established, the next step is to determine how the network expresses its prediction of the heliostat surface. Since the choice of surface representation directly affects both prediction accuracy and computational efficiency, this work adopts the NURBS-based format introduced in Section 2.3.1, which offers a compact yet flexible description of complex geometries. In the implementation used here, the x- and y-coordinates of these control points are fixed to a predefined grid, leaving the network to predict only the z-coordinates. For a four-facet heliostat, this corresponds to 8×8 z-values per facet, resulting in 256 predicted values per surface.

A key challenge when predicting NURBS control points from flux density images is the ill-posed nature of the inverse problem: multiple control point configurations can yield indistinguishable optical outputs. To evaluate this ambiguity, the predicted control points are compared with the ground truth control points of the surfaces used to generate the input images during training. This enables quantification of the deviation between optical accuracy and geometric fidelity.

Once predicted, the NURBS control points — consisting of the fixed x-y grid and the predicted z-coordinates — are passed to the differentiable ray tracer. The ray tracer then produces the corresponding flux density image, integrating a physics-based step into the feedback loop for network optimization.

4.1.4. Raytracing

The ARTIST differentiable ray tracer (Section 2.3.1) is employed both during dataset generation and within the training loop. In both contexts, the available parallelization strongly influenced the design of this work's pipeline.

Data generation

When constructing the training and test datasets, surfaces must be ray traced under multiple Sun positions. The ray tracer supports parallel evaluation of several heliostats, but only if the Sun position is identical across them (Section 2.3.1). This restriction arises because the ray-tracing scene must be rebuilt whenever the Sun position changes. To exploit parallelism, flux density images were therefore generated in groups of surfaces sharing the same Sun positions. This greatly reduced dataset generation time but imposed a structural constraint on how data could later be batched.

Training loop

During training, the network predicts batches of heliostat surfaces that must each be converted into flux density images for comparison with reference images. To keep runtimes manageable, these surfaces are ray traced in parallel. However, parallelization is only possible if all surfaces in a batch share the same Sun position. As a result, each batch is evaluated under a single Sun position, even though the training data contain multiple Sun positions per heliostat. Evaluating all available Sun positions per batch would require repeating the ray tracing step for each one, increasing runtime by a factor equal to the number of Sun positions. The adopted strategy therefore trades off feedback richness—fewer supervisory signals per iteration—for substantially improved computational efficiency.

In addition, this parallelization constraint structures the data into fixed chunks defined by shared Sun positions. Randomization during data loading is therefore limited to shuffling within these chunks, rather

than across the entire dataset. This reduces training diversity slightly but was judged acceptable given the significant runtime benefits.

Trade-offs
This design creates a trade-off between computational efficiency and data diversity. On the one hand, large batches maximize parallelism but reduce the variability of training data. On the other hand, small batches increase stochasticity but diminish throughput, as the maximum parallelization scales directly with the batch size. A compromise was therefore chosen by fixing batches at eight surfaces under shared Sun positions, balancing runtime efficiency against training diversity.

4.1.5. Loss Calculation

In the training loop, the ray tracer serves as the bridge between the predicted heliostat surface and the optical output in the form of a flux density image. To enable the network to learn from this output, the generated image is compared to the ground truth image associated with the same heliostat–Sun position pair. This comparison is performed numerically using an image loss function, which converts the pixel-wise differences into a single scalar value that guides the weight updates of the neural network.

The choice of loss function is non-trivial. Different metrics emphasize different aspects of image similarity: per-pixel measures such as the Mean Absolute Error (MAE), Mean Squared Error (MSE) and the Root Mean Squared Error (RMSE) are sensitive to intensity differences, while the Structural Similarity Index Measure (SSIM) highlights spatial structure and contrast. Consequently, the loss function must be selected with care, balancing robustness, sensitivity, and comparability with prior work.

Following Lewen et al. [24], MAE is adopted as the default image loss function in this work. Preliminary experiments conducted with MAE, MSE, RMSE, and SSIM (see Appendix A.6) confirmed that while SSIM can provide slightly better flux-image similarity, MAE yields more consistent surface reconstructions. A single SSIM experiment will additionally be run among the final experiments to confirm this choice.

With the image loss function defined, the training loop has a mechanism to update weights during learning. However, the quantity minimized during training is not identical to the quantities of interest when judging the model performance in this case. To objectively compare models and quantify surface reconstruction quality, explicit performance metrics must be established.

4.1.6. General Performance Metrics

Two primary metrics are employed to evaluate performance of the trained model.

1. **Flux MAE:** the MAE between the predicted and ground truth flux density images. This quantifies how well the network reproduces the reference optical output, which is the indirect but original observable in the inverse problem.
2. **Surface MAE:** the MAE between the predicted and reference surfaces. This is computed as the average MAE across all z-coordinates of the predicted and ground truth NURBS control points. The resulting single scalar value provides a direct measure of the geometric accuracy of the surface reconstruction.

Each trained model is evaluated on the same test set as Lewen et al. to enable comparability, and both metrics are recorded to capture complementary aspects of performance: flux reproduction and surface reconstruction.

In addition to these MAE-based metrics, the relative flux density accuracy metric defined by Lewen et al.(Section 2.3.1) is also employed for direct comparison. A practical challenge arises because this work applies max-normalization during training, ensuring all flux density bitmap pixel values lie in the range $[0, 1]$. For evaluation, however, flux images are sum-normalized to match Lewen et al.'s convention, so that the overall accuracy can be computed consistently as

$$ACC = \sum \left| \phi_{\mathrm{GT}} - \phi_{\mathrm{pred}} \right| . \tag{4.3}$$

With both training losses and performance metrics now defined, the framework can quantify reconstruction performance. Yet, because the inverse problem remains strongly ill-posed, good flux density

prediction does not guarantee physically accurate surfaces. To address this, the following section introduces geometric descriptors derived from real heliostat measurements, which form the basis for designing regularization strategies.

4.1.7. Geometric Descriptors of Heliostat Surfaces

Analyzing real or predicted heliostat surfaces requires a consistent set of quantitative measures that capture their key geometric properties. Without such definitions, quantities such as curvature, tilt, or edge behavior would remain ambiguous, making it difficult to interpret measurements or link them to physical properties.

To address this, a set of geometric descriptors is defined. These descriptors serve two purposes: first, they enable the empirical analysis of deflectometry data in Section 4.1.8; second, they provide the basis for the regularization terms formulated later in Section 4.1.9. The following definitions establish the mathematical framework used throughout the remainder of this work.

z-Values

Let $Z \in \mathbb{R}^{4\times8\times8}$ denote the z-coordinates of the control points for the four facets of a heliostat, expressed relative to an ideal reference surface at $z = 0$. These values represent deviations from the ideal geometry after removing canting effects. Basic statistics of Z—such as mean, range, and standard deviation—capture the overall magnitude of deviations.

Curvature

Local smoothness is quantified using a discrete Laplacian operator, which approximates the second-order spatial derivative. For a control point (i, j) on facet f, the Laplacian is computed as:

$$\Delta Z_{i,j}^{(f)} = Z_{i-1,j}^{(f)} + Z_{i+1,j}^{(f)} + Z_{i,j-1}^{(f)} + Z_{i,j+1}^{(f)} - 4Z_{i,j}^{(f)} \tag{4.4}$$

This quantity measures the deviation of $Z_{i,j}^{(f)}$ from the average of its four immediate neighbors. The mean curvature of a surface is defined as the average absolute Laplacian over all control points and facets. Units are in $1/\mathrm{m}$.

Tilt

Global orientation is described by the unit normal vector of each facet. The normal n_{bf} for facet f in batch b is obtained by fitting a plane to its control points via singular value decomposition (SVD). The tilt angle relative to the ideal z-axis is then:

$$\theta_{bf} = \arccos\left(\left| n_{bf} \cdot \begin{bmatrix}0\\0\\1\end{bmatrix} \right|\right) \tag{4.5}$$

A tilt of $\theta_{bf} = 0$ corresponds to perfect alignment with the z-axis.

Edge Dip

To capture localized bending at facet boundaries, the height difference between each edge control point Z_{edge} and its adjacent inner neighbor Z_{inner} is defined as:

$$\delta Z = Z_{\mathrm{edge}} - Z_{\mathrm{inner}} \tag{4.6}$$

Negative values indicate that the edge lies below the interior, as frequently observed in real measurements.

These geometric descriptors provide interpretable, physically meaningful measures of heliostat surface geometry. They serve as the foundation for both the empirical analysis of real heliostat data (Section 4.1.8) and the formulation of regularization strategies that incorporate physical constraints (Section 4.1.9).

4.1.8. Empirical Constraints from Deflectometry Measurements

Regularization strategies should reflect the physical characteristics of real heliostat surfaces. However, without quantitative knowledge of these characteristics, constraints risk being overly strict, too loose, or misaligned with actual surface behavior.

To address this, this section analyzes measurements obtained via deflectometry at the CSP plant in Jülich, using the geometric descriptors defined in Section 4.1.7 to establish empirical bounds. These bounds later inform the mathematical constraints described in Section 4.1.9. The measurements represent surface deviations with canting removed by subtracting the corresponding ideal heliostat from the same position (including identical canting values), as described in Section 4.1.1, resulting in data that captures only deviations from the ideal geometry on a constant xy-grid.

z-Values Distribution

Understanding the order of magnitude of surface deviations is a prerequisite for designing realistic constraints. To quantify these deviations, the statistical distribution of the z-coordinates was computed across all measured surfaces. Statistics were calculated over all z-coordinates—$4 \times 8 \times 8$ per heliostat—for all 423 heliostats, yielding 108 288 individual data points.

The results, shown in Table A.5 indicate that both the mean and absolute mean are on the order of 10^{-5} m to 10^{-4} m, confirming that real surface deviations are comparatively small. This supports the later introduction of an L2 regularization term to encourage predictions to remain close to zero.

The distribution of z-coordinates, shown in Figure 4.6, reveals a strong concentration around the mean, with relatively few extreme values. As most coordinates fall within ± 2 mm, a range-based regularization will also be implemented to penalize predictions outside this interval while allowing occasional outliers to capture rare but physically plausible deviations.

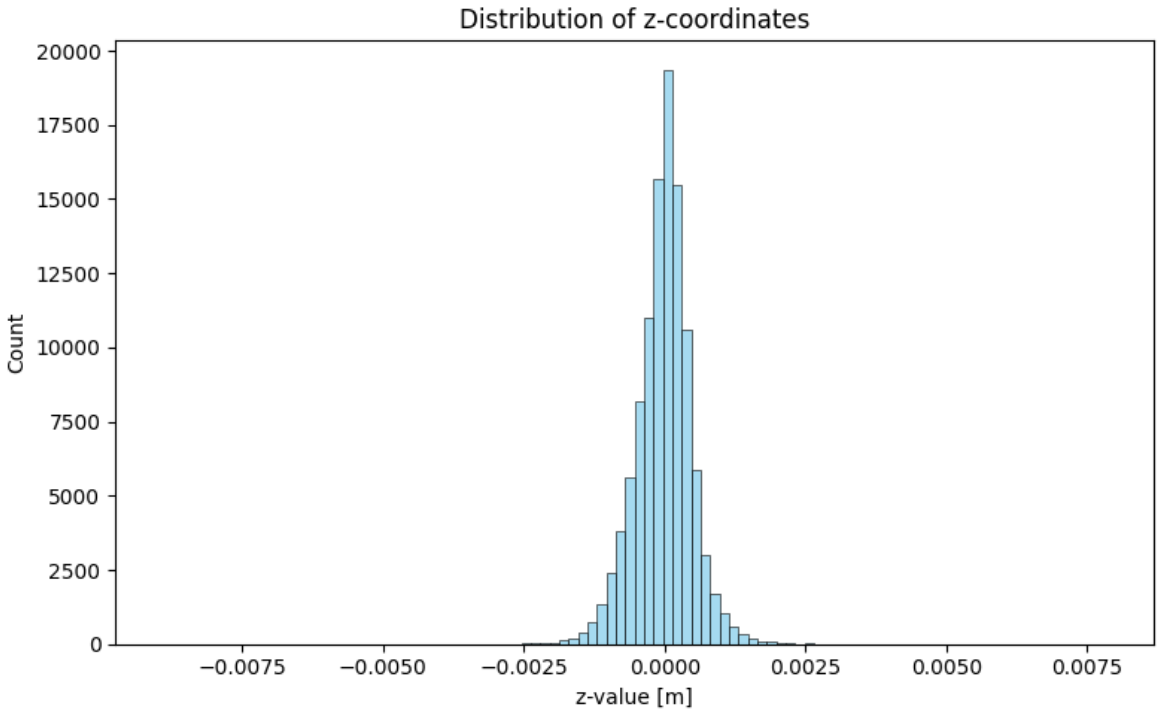


Figure 4.6: Statistical distribution of z-coordinate deviations across all measured heliostat surfaces. The majority of values are concentrated tightly around zero, within approximately ± 2.5 mm, indicating that real surface deviations are comparatively small. This distribution supports the use of regularization strategies that penalize large deviations while still allowing for occasional outliers.

Beyond the range and distribution of individual z-values, the overall surface shape and structural patterns are also relevant, as they may inform the design of spatially aware regularization terms. The

next subsections extend this analysis to curvature, tilt, and edge behavior, providing surface-shape constraints.

General Surface Geometry

While the statistical analysis of individual z-coordinates describes the magnitude of surface deviations, it does not reveal how these deviations are distributed across the entire heliostat. Understanding the overall surface shape is important, as recurring geometric patterns can influence the choice of regularization strategies.

To address this, representative measured surfaces were plotted for visual inspection, enabling the identification of common structural characteristics. Figure 4.7 shows an example of a typical surface from the dataset, illustrating patterns that will be quantified in the following subsections.

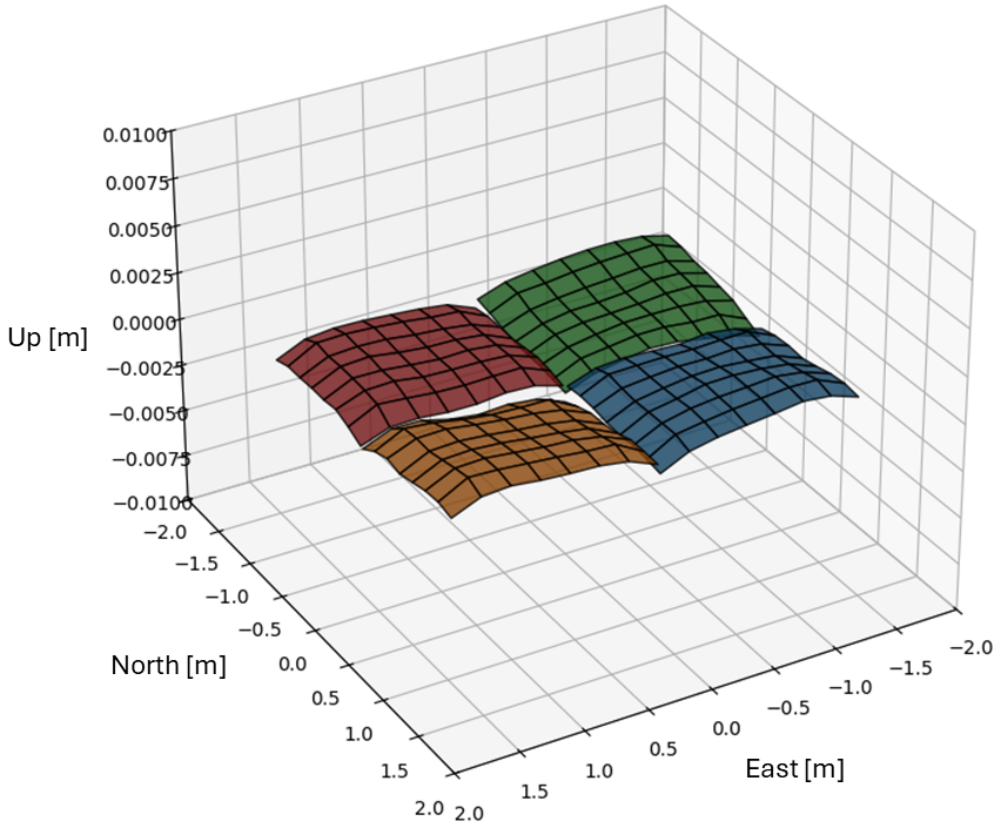


Figure 4.7: Example of a representative measured heliostat surface (AA33) from the Jülich dataset. The surface is composed of four flat rectangular facets, each exhibiting small but noticeable deviations from the ideal, flat plane. Such patterns provide insight into common geometric characteristics of real heliostats and form the basis for the quantitative analysis of surface errors in the following subsections.

Three key observations emerge. First, the surfaces display a high degree of smoothness, with stronger curvature typically occurring near the edges. Second, it seems that even without canting, some of the surfaces are slightly tilted. Third, the control points along the edges consistently dip below their immediate inner neighbors.

The following subsections quantify these qualitative observations by analyzing different aspects of surface geometry:

- **Curvature**, to capture overall smoothness patterns.
- **Tilt**, to measure directional inclination of facets.
- **Edge behavior**, to assess the consistent dip of boundary control points.

Each of these analyses provides parameters and empirical ranges that later guide the implementation of the corresponding regularization terms.

Curvature Distribution While z-coordinate statistics quantify the magnitude of deviations, they do not capture local surface roughness. Curvature provides this additional perspective, revealing how smoothly or abruptly the surface changes between neighboring control points. High local curvature may indicate unrealistic surface shapes that should be constrained during prediction.

The curvature for each control point was computed as defined in Section 4.1.7. Table A.6 summarizes the resulting statistics, based on the same 108 288 control points as in the z-coordinate analysis.

The distribution of curvature values (Figure 4.8) shows three distinct peaks: values near 0 (center points), around 0.04 (edge points), and around 0.08 1/m (corner points). This indicates that most large curvatures occur along the facet boundaries, while the centers remain nearly flat.

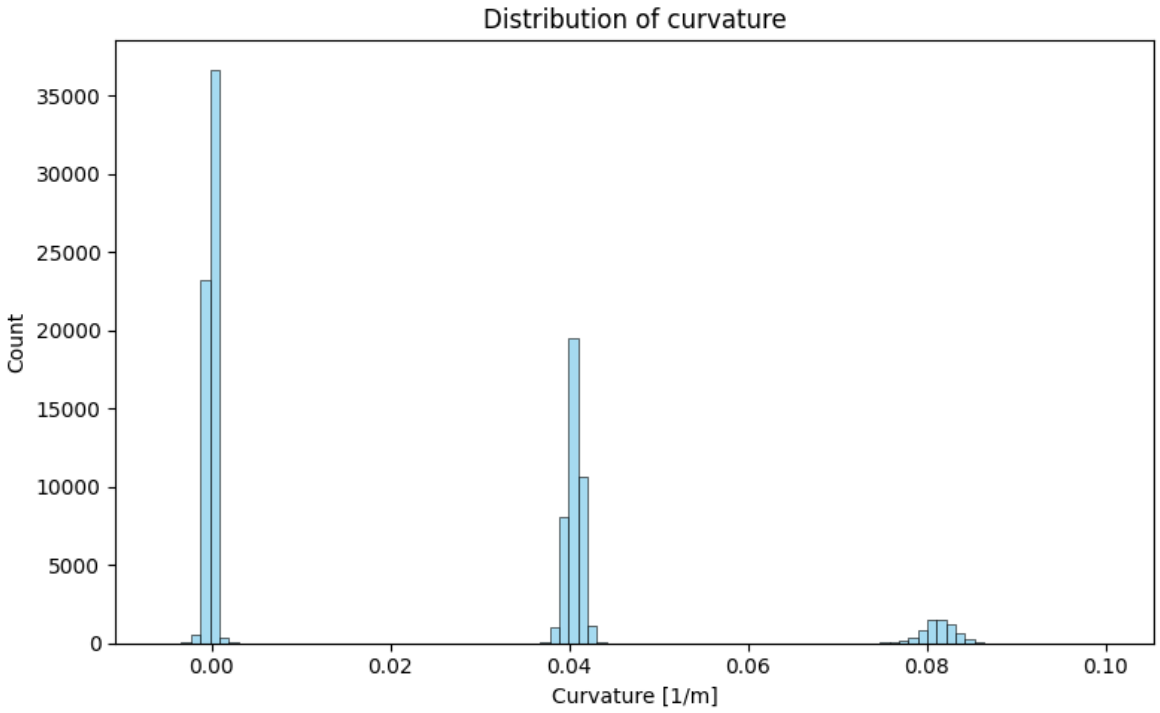


Figure 4.8: Distribution of curvature values across all heliostat control points. Three distinct peaks are visible: values near 0 1/m corresponding to center points, around 0.04 1/m for edge points, and around 0.08 1/m for corner points.

To confirm this, the control points were split into edge and center regions. The results confirm that edges account for nearly all significant curvature values, while the centers remain close to zero. Detailed statistics are provided in Table A.7.

Further inspection of the highest-curvature surfaces shows that the extreme values all occur at facet corners. This is partly an artifact of the Laplacian calculation, which uses zero-padding for points outside the facet, thereby exaggerating curvature values at the boundaries and particularly at the corners. However, this concentration of high curvature at the edges is also physically plausible: the material at the facet boundaries has more degrees of freedom and is less constrained, which can lead to larger deviations than in the more rigid central region. Since the Laplacian artifact and the genuine physical effect are difficult to disentangle, curvature regularization will be applied only to center points to avoid penalizing artificial edge exaggerations while still capturing realistic surface smoothness.

The highest curvature value is observed on the AA36 heliostat in the Jülich field, as shown on the left in Figure 4.9. This surface appears to exhibit an issue, as the facets seem to include a canting angle, especially when compared to Figure 4.7. A possible explanation is that two heliostats were swapped

during measurement, causing the canting to be computed incorrectly. Since this data originates directly from the PAINT database, it is however impossible to find the definite origin of this error. Nevertheless, the surface is retained in the generation and augmentation of the training data. In machine learning applications, edge cases often provide valuable information, so this surface was not excluded despite being visibly different from the other surfaces [43].

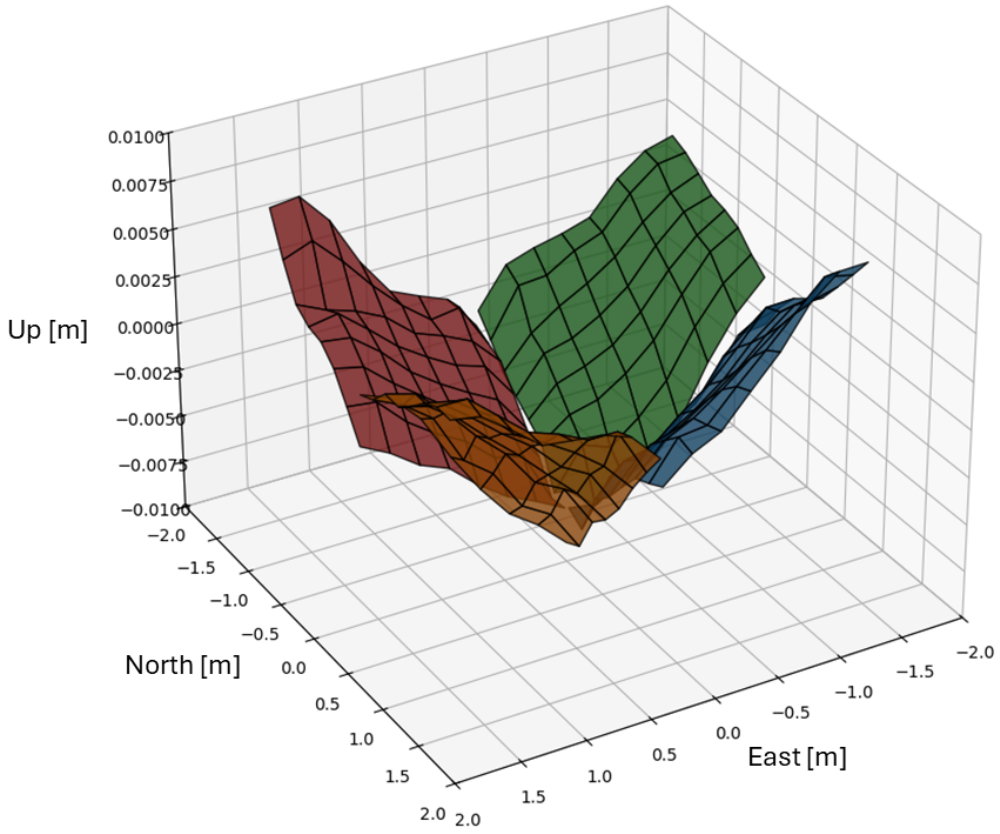


Figure 4.9: Example of a heliostat surface (AA36) with the highest measured curvature values in the Jülich field. The surface exhibits irregular canting, likely due to measurement inconsistencies or swapped heliostats.

The reference curvature for regularization was set to the mean curvature of the center points (0.000281 1/m). An initial deviation threshold equal to this mean was used, but later reduced to 0.00005 1/m after preliminary tests indicated that the original threshold insufficiently penalized high-curvature predictions.

In addition to curvature, the following subsection examines the global tilt of the facets to determine whether a tilt regularization term is warranted.

Tilt Distribution While curvature describes local surface roughness, tilt measures the global orientation of each facet. Excessive tilt could indicate a systematic misalignment, even if the surface is otherwise smooth.

Tilt angles relative to the vertical normal vector were computed as defined in Section 4.1.7. The results in Table A.8 show that the mean observed tilt is $0.043°$, or 0.76 mrad. Such small values confirm that the measured surfaces are effectively flat in terms of overall orientation, without any excessive tilt.

Given this, tilt regularization will be configured to penalize any significant deviation from the vertical normal vector, ensuring that predicted facets remain globally aligned while allowing for the minor variations observed in the real data.

Lastly, the behavior of the surface edges is analyzed, as they display distinctive structural patterns.

Edge Dip Distribution Visual inspection of measured surfaces (e.g., Figure 4.7) shows that edge control points generally have lower z-coordinates than their adjacent inner points. This effect is most pronounced at the corners, where two neighboring points are missing. Such dips are likely caused by the NURBS surface representation and the lack of geometric constraints at the boundaries, rather than by the physical properties of the real heliostats.

Nevertheless, this characteristic is consistently present in the parameterized NURBS surfaces used to generate the flux density images. To ensure consistency between the training data and predicted surfaces, this behavior is modeled explicitly through an edge dip regularization. The edge dip is defined as the z-coordinate difference between an edge point and its immediate inner neighbor, as described in Section 4.1.7. Table A.9 summarizes the observed statistics; the mean dip value is used as the margin parameter for the regularization term.

The deflectometry analysis therefore provides quantitative parameters for all surface-shape regularizations: curvature thresholds, edge dip margins, and z-value constraints. These parameters are applied during training to guide the network toward realistic and physically consistent predictions. Building on these empirical constraints, the following section describes how they are implemented mathematically to reduce the solution space and prevent the network from converging to implausible geometries.

4.1.9. Further Reduction of Solution Space

Predicting a heliostat surface from flux density measurements is an inherently ill-posed inverse problem: many distinct geometries can produce nearly identical flux density images. Without additional constraints, the neural network may converge to a surface that matches the flux density but is geometrically unrealistic.

The empirical bounds derived from deflectometry analysis (Section 4.1.8) provide a quantitative definition of realistic surface shapes. These bounds form the basis for the geometry-based constraints, which are implemented as penalty terms in the training loss to limit surface magnitude, enforce smoothness, encode known structural patterns, and promote prediction diversity.

In addition to geometry-based constraints, data-level strategies can further reduce ambiguity by altering heliostat positions to increase the information content of the flux density images. This complementary approach aims to make the inverse problem better conditioned.

The following subsections first present the geometry-based constraints—directly linked to the physical measurements—before introducing the data-level strategies.

z-Values Regularization

The first geometry-based regularization strategy constrains the z-coordinate of the predicted control points. This constraint is directly motivated by the deflectometry analysis in Section 4.1.8, which showed that real heliostat surface deviations are small—typically within a few millimeters.

Without such a bound, the neural network may produce unrealistically large deviations during early training. In extreme cases, these exaggerated shapes can cause the ray tracer to output completely black flux density images. Once this occurs, the image loss may stagnate, and the model can converge prematurely to an incorrect solution.

To prevent this, the absolute z-coordinates of all predicted control points are limited to a maximum threshold z_{lim}, set according to the measured data. A penalty term is added to the loss whenever this threshold is exceeded, proportional to the violation magnitude and scaled by a high penalty factor $\gamma_{\text{z-lim}}$:

$$L_{\text{z-lim,reg}} = \gamma_{\text{z-lim}} L_{\text{z-lim}} \tag{4.7}$$

Here, $L_{\text{z-lim}}$ represents the violation magnitude beyond z_{lim}.

While the z-limit enforces a realistic range, it does not actively encourage small deviations. Therefore, the next subsection introduces an L2 regularization term to bias the network toward minimal but physically plausible departures from the ideal heliostat surface.

L2 Regularization
The second geometry-based constraint builds on the deflectometry findings in Section 4.1.8, which showed that real heliostat surfaces deviate from the ideal shape only slightly—typically on the order of millimeters or less. This motivates biasing the predicted z-coordinates toward small values that reflect physically plausible geometries.

Even with the z-limit regularization in place (Section 4.1.9), the network may still produce unnecessarily large variations within the allowed range. Such shapes can be unrealistic yet still yield low image loss because of the ill-posed nature of the problem.

To counter this, an L2 regularization term is applied directly to the predicted z-coordinates:

$$L_{L2} = \|Z\|_2^2 \tag{4.8}$$

Here, Z is the $4 \times 8 \times 8$ tensor of predicted z-coordinates, and $\|Z\|_2^2$ denotes its squared L2 norm:

$$\|Z\|_2^2 = \sum_{f=1}^{4} \sum_{i=1}^{8} \sum_{j=1}^{8} Z_{i,j,k}^2 \tag{4.9}$$

Where f indexes the four facets, and i, j denote the 8×8 control points. A weight factor γ_{L2} determines the impact of this regularization on the overall loss:

$$L_{\text{L2,reg}} = \gamma_{\text{L2}} L_{\text{L2}} \tag{4.10}$$

The influence of $L_{\text{L2,reg}}$ can be scheduled over the course of training. A high initial penalty helps keep predictions close to the ideal surface during early training, aiding stable convergence. Later, the penalty can be reduced to allow more flexibility for refining the predicted geometry. If the penalty remains too strong throughout, the model risks producing overly flat surfaces that fail to capture meaningful deviations. In this work, primarily linear decay of the loss term is investigated.

While L2 regularization encourages small deviations globally, it does not guarantee local smoothness. To address this, the next section introduces a curvature-based regularization that penalizes excessive local roughness in the predicted surfaces.

Curvature Regularization
The third geometry-based constraint addresses local smoothness. Deflectometry analysis (Section 4.1.8) showed that real heliostat surfaces are generally smooth, with higher curvature only near the edges and minimal abrupt variations in the center. This motivates a regularization term that penalizes excessive local roughness in the predicted geometry.

Even when z-limits (Section 4.1.9) and L2 regularization are applied, the network may still generate surfaces that are globally plausible but locally jagged. Such oscillations can still achieve low image loss because multiple shapes can produce nearly identical flux density images.

The curvature at each control point is computed as defined in Section 4.1.7 using the discrete Laplacian operator. For each predicted surface, the mean curvature is compared to a reference value C_{ref} obtained from the real-surface analysis. A loss is applied only when the deviation exceeds a tolerance τ:

$$L_{\text{curve}} = \begin{cases} \frac{1}{N} \sum_{i \in \mathcal{I}} \left(C_{\text{pred},i} - C_{\text{ref}}\right)^2 & \text{if } \mathcal{I} \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \quad \text{where } \mathcal{I} = \{i \mid |C_{\text{pred},i} - C_{\text{ref}}| > \tau\} \tag{4.11}$$

where $C_{\text{pred},i}$ is the predicted curvature of sample i, C_{ref} is the reference curvature, τ is the curvature deviation threshold, $\mathcal{I}$ is the set of indices where the deviation exceeds τ, and N is the number of samples in $\mathcal{I}$. The loss is scaled by a weight factor γ_{curve}:

$$L_{\text{curve,reg}} = \gamma_{\text{curve}} L_{\text{curve}} \tag{4.12}$$

This regularization is activated only after a warm-up period of several epochs. Applying it too early could destabilize training because initial predictions may be highly curved, leading to large penalties and poor convergence. Starting later allows the network to establish the general shape before fine-tuning smoothness.

While curvature regularization enforces local smoothness, it does not address potential global misalignments of an entire facet. Even a uniformly tilted surface can produce smooth curvature values while deviating from the correct orientation. The next subsection therefore introduces a tilt regularization to maintain proper global alignment with respect to the vertical axis.

Tilt Regularization
Even when constraints on magnitude, smoothness, and edge shape are satisfied, the predicted surface can still exhibit a uniform tilt relative to the z-axis. Such global misalignments may have little effect on the flux density image, but they reduce the physical realism of the reconstructed geometry.

To counter this, a tilt regularization term penalizes deviations of each facet's normal vector from the vertical, as defined in Section 4.1.7. The contribution of this term to the overall loss is scaled by a weight factor:

$$L_{\text{tilt,reg}} = \gamma_{\text{tilt}} L_{\text{tilt}} \tag{4.13}$$

This encourages the network to maintain correct global facet orientation in addition to producing locally plausible surface shapes.

While tilt regularization enforces correct global orientation, it does not address localized structural patterns observed in real surfaces. One such pattern is the systematic dip of control points along facet edges, which is modeled by the edge dip regularization introduced in the next subsection.

Edge Dip Regularization
Deflectometry measurements reveal a recurring local feature in real heliostat facets: the outermost control points, located along the first and last rows and columns, tend to be slightly lower in the z-direction than their immediate inner neighbors (Figure 4.10). This bending effect occurs independently of canting.

General smoothness constraints, such as curvature regularization (Section 4.1.9), treat all points uniformly and cannot enforce this localized deformation. Without an explicit constraint, the network may produce edges that are unrealistically flat or even raised relative to the interior, contradicting real-surface behavior.

To encode this feature, an edge dip regularization is introduced. Let Z_{edge} be the z-coordinate of an edge control point and Z_{inner} that of the immediately adjacent inner control point. A margin m, derived from statistical analysis, specifies the expected drop in z-values between the inner and outer points. All distances are measured in meters.

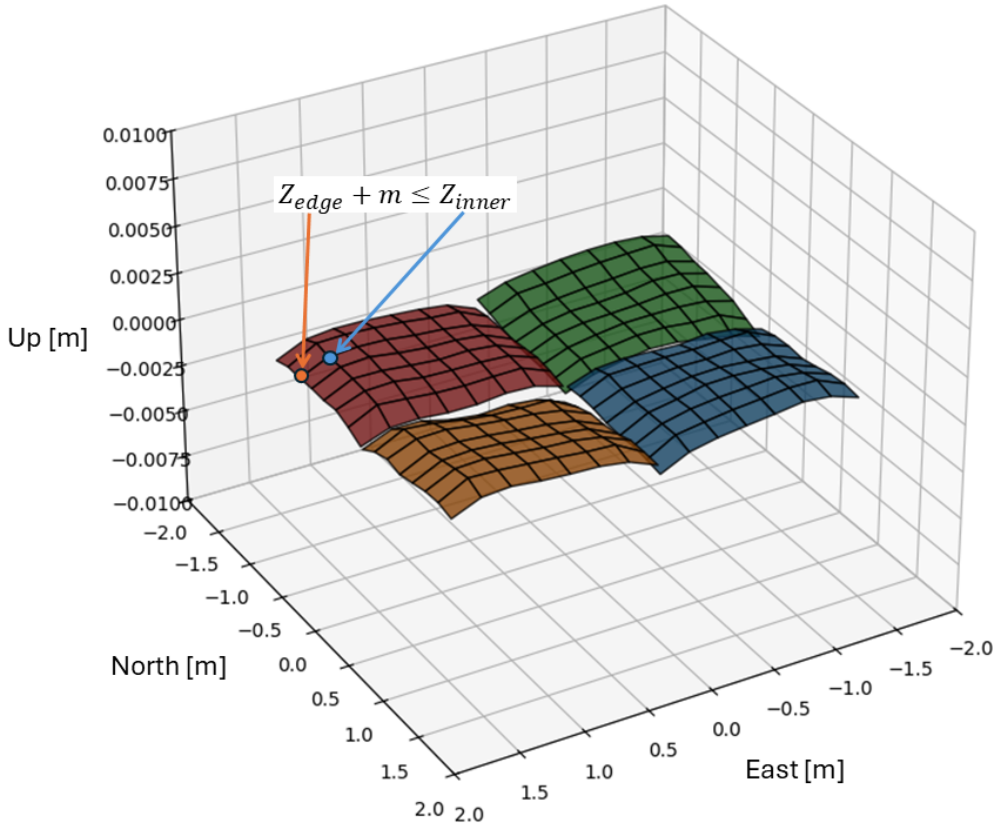


Figure 4.10: Visualization of the edge dip effect, showing the systematic lowering of the outermost facet control points (Z_{edge}) relative to their immediately adjacent inner points (Z_{inner}). This feature, consistently observed in measured heliostats, is incorporated into the edge dip regularization term to better reflect realistic surface behavior.

The constraint is expressed as:

$$\Delta = Z_{\text{edge}} - Z_{\text{inner}} + m \leq 0 \tag{4.14}$$

where a penalty is applied whenever $\Delta > 0$. The corresponding loss term is defined as:

$$L_{\text{dip}} = \frac{1}{4}\left(E_{i,j}[\Delta^2_{\text{top}}] + E_{i,j}[\Delta^2_{\text{left}}] + E_{i,j}[\Delta^2_{\text{right}}] + E_{i,j}[\Delta^2_{\text{bottom}}]\right) \tag{4.15}$$

with $E_{i,j}[\cdot]$ denoting the mean along a given edge, and Δ^2_{top} refers to the squared violation for the top row, as defined in Equation 4.14. The final regularization term added to the training loss is:

$$L_{\text{dip,reg}} = \gamma_{\text{dip}} L_{\text{dip}} \tag{4.16}$$

where γ_{dip} is the penalty factor that controls the strength of this regularization.

By incorporating this term, the network learns to reproduce realistic edge bending patterns, complementing the broader smoothness constraints of curvature regularization.

Together, curvature, tilt, and edge dip constraints guide each facet toward shapes consistent with real heliostat surfaces. However, these terms alone do not guarantee variety in the solutions: the network may still settle on a single, suboptimal geometry that satisfies all constraints. The next subsection presents a repulsion regularization that promotes exploration of alternative, physically valid surface shapes.

Repulsion Regularization
Even with the described geometric regularizations, the inverse problem remains ill-posed: many different surface geometries can produce nearly identical flux density images. As a result, the network may converge to a surface that matches the flux density but is geometrically incorrect, limiting the exploration of other plausible shapes.

This lack of diversity in predictions reduces the likelihood of discovering the true surface geometry. Once the model finds a shape that yields a sufficiently low image loss, it may repeatedly converge to this geometry, ignoring alternative solutions that could match the optical data more accurately.

To address this, a repulsion regularization strategy is introduced. The method stores previously predicted surfaces in a buffer. If the network generates a new prediction that is too similar to any stored surface, a penalty is applied to the loss. This forces the network to search for geometrically distinct shapes that still match the flux density images. Training continues over many epochs, and the final model is chosen as the one achieving the lowest surface loss on the test set, increasing the chances of capturing the ground truth surface among the explored alternatives.

Formally, the repulsion loss is defined as:

$$\mathcal{L}_{\text{rep}} = \frac{1}{N} \sum_{i=1}^{N} \left(\cos(\theta_i)\right)^2 \tag{4.17}$$

where L_{rep} is the repulsion loss, N is the number of stored surfaces, and $\cos(\theta_i)$ is the cosine similarity between the current prediction and the i-th stored surface. Cosine similarity is chosen because it emphasizes differences in shape rather than magnitude:

$$\cos(\theta_i) = \frac{\mathbf{z}_{\text{current}} \cdot \mathbf{z}_{\text{i, past}}}{\|\mathbf{z}_{\text{current}}\| \cdot \|\mathbf{z}_i\|} \tag{4.18}$$

where $\mathbf{z}$current is the current surface prediction and $\mathbf{z}i, \text{past}$ is the i-th stored surface in the buffer. The final repulsion term added to the training loss is:

$$L_{\text{rep,reg}} = \gamma_{\text{rep}} L_{\text{rep}} \tag{4.19}$$

where γ_{rep} controls the strength of this penalty.

The regularizations described so far operate directly on the predicted geometry, constraining its magnitude, smoothness, orientation, and diversity. However, the ill-posed nature of the problem can also be addressed at the data level by increasing the information content of the flux density images. The following strategy modifies the training data itself—specifically, the heliostat positions—to generate flux images that are more sensitive to surface variations, providing the network with stronger cues for reconstruction.

Synthetic Heliostat Positions
The regularizations described so far constrain the predicted geometry. An alternative way to reduce the ill-posedness of the inverse problem is to increase the information content of the input data itself. In this context, the spatial configuration of heliostats plays a crucial role in shaping the flux density patterns that the network learns from.

While the real heliostat positions from the Jülich CSP plant are used for most experiments, alternative synthetic layouts are explored to assess how geometric configuration affects prediction performance. The working hypothesis is that heliostats located closer to the receiver generate sharper, more localized flux patterns. A narrower angular spread of reflected sunlight should yield features with higher spatial contrast, potentially making the inverse problem less ambiguous.

In the real plant, heliostats are located tens to hundreds of meters from the receiver, producing more diffuse flux distributions and exacerbating ambiguities in surface reconstruction. To test the effect of

optical sharpness, synthetic datasets are generated with heliostats placed at random distances of 5–15 m and 30–70 m from the receiver. Here, the distance is the direct line-of-sight from the heliostat center to the receiver. Because the Jülich receiver is mounted more than 50 m above ground level, these configurations are physically unrealistic—the simulated heliostats would effectively "float" in mid-air. Figure 4.11 illustrates a random layout within the 30–70 m range on the left and a slightly different layout for the 5-15 m experiments on the right.

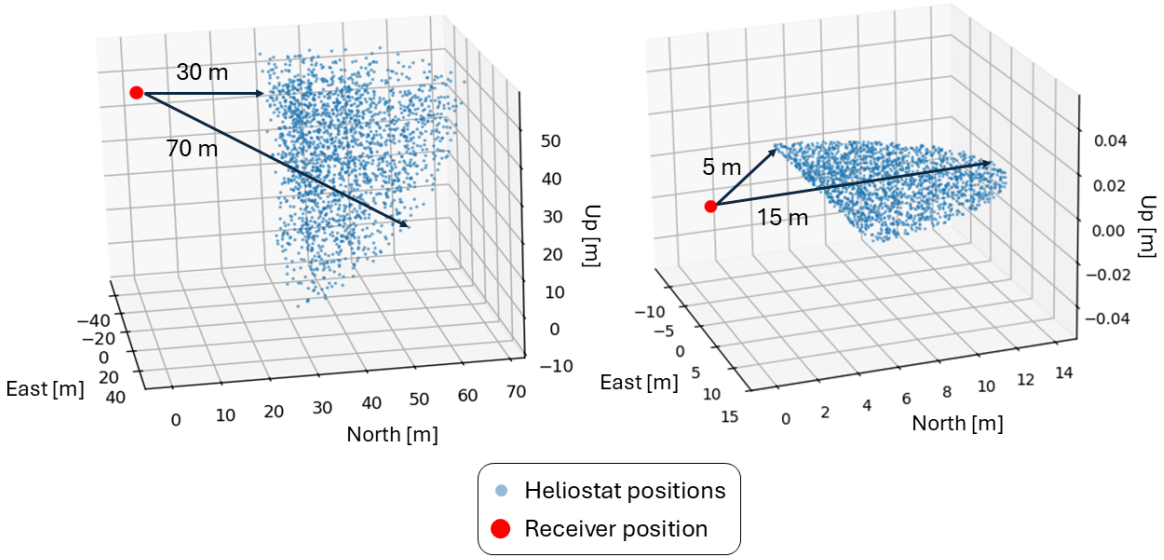


Figure 4.11: Randomly generated heliostat layouts used in synthetic datasets. Left: positions sampled within a spherical shell spanning 30–70 m from the receiver. Right: positions sampled within a 5–15 m range, constrained to a horizontal plane at receiver height to approximate a flat-field arrangement. The receiver is marked in red, and heliostat positions in blue.

For the 5–15 m dataset, a slightly different heliostat position scheme was adopted. Instead of sampling from a three-dimensional shell, the positions were drawn randomly from a horizontal plane at the same height z as the receiver. This adjustment was motivated by providing a closer conceptual approximation to real heliostat fields, which are typically arranged on (nearly) flat ground. These positions are illustrated by Figure 4.11. This means that for these experiments, the receiver is also in the same plane as the heliostats.

The primary aim of these additional synthetic datasets is to evaluate whether such adaptations reduce ill-posedness and improve prediction accuracy. If beneficial, this could enable a two-stage training strategy: pretraining on close-distance configurations to learn the core shape–flux mapping, followed by fine-tuning on the real, more complex field geometry.

However, this modification of heliostat positions introduces a secondary challenge. Canting vectors, which define the facet orientations for focusing sunlight onto the receiver, are inherently linked to real-world heliostat positions. Recomputing canting for synthetic positions would require additional optical design steps. Instead, the canting vectors from the original Jülich layout are retained, and only the heliostat positions are replaced with synthetic ones. This mismatch leads to flux density images such as the one shown in Figure 4.12, where the reflected light from each facet is no longer co-focused but appears as separate bright spots.

Interestingly, this unintended defocus may provide an advantage: the individual facet reflections can make certain surface features more visible in the flux density images, potentially reducing ambiguity in the inverse mapping. To explore this further, an additional experiment is conducted in which the canting vectors are explicitly included as an extra neural network input, allowing an evaluation of whether supplying this information improves prediction quality.

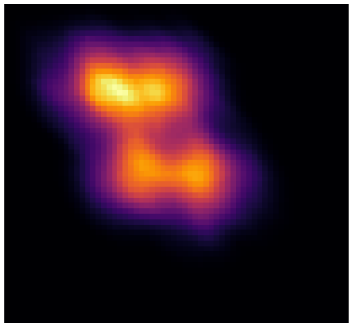


Figure 4.12: Example flux density image of a heliostat with mismatched canting, where reflected light from individual facets is not co-focused, producing separate bright spots on the receiver.

Summary of Regularization Strategy
The regularization approach developed in this work explores both geometry based regularizations as well as dataset-level modifications. Each component targets a specific aspect of the prediction task, and together they address both the physical realism of individual surfaces and the broader challenge of exploring the solution space. They were introduced in the following order:

- **z-limit regularization:** Prevents unrealistic surface magnitudes.
- **L2 regularization:** Encourages small deviations from the ideal shape.
- **Curvature regularization:** Enforces smoothness of the surface.
- **Tilt regularization:** Aligns global surface orientation with the expected z-direction.
- **Edge dip regularization:** Models the natural bending of facet edges.
- **Repulsion regularization:** Promotes diversity and mitigates convergence to suboptimal local minima.
- **Dataset-based strategy:** Uses synthetic heliostat positions closer to the receiver to produce sharper flux images and reduce problem ill-posedness.

Together, these constraints guide the network toward physically plausible, varied, and accurate surface reconstructions while also encouraging broader exploration of the solution space.

4.1.10. Regularization Implementation

Having established the set of regularizations used in this work, the next step is to integrate them into the training pipeline in a way that allows for controlled activation, tuning, and comparison. This section describes how each regularization was implemented, how its parameters were selected, and how two successive configurations—one manually designed and one obtained via Bayesian optimization—were developed.

The regularizations described in Section 4.1.9 will be integrated into the training loop as modular components, each of which could be activated or deactivated independently. This design makes it possible to isolate their individual effects and to explore different combinations during experimentation.

Selection of Regularization Parameters
Initial parameter estimates for each regularization are derived from the deflectometry-based surface analysis presented in Section 4.1.8. These values serve as physically motivated starting points, which are then refined through local test runs-with a low number of training samples-to evaluate their effect on convergence and stability.

For each active regularization, the weight factor is tuned so that its contribution represented roughly 1–20% of the total loss. This range ensures that the regularization influences optimization without overwhelming the image loss.

However, two exceptions are made: The *z-limit* regularization and the *regularization on outputs* receive significantly higher initial weights, as both play a decisive role in constraining predicted magnitudes

early in training. For the z-limit, the weight remains constant to strictly enforce predefined boundaries. For the regularization on outputs, the weight starts high and decays linearly, ensuring a strong early influence that diminishes once other regularizations became dominant.

Two main configurations of regularization parameters are tested:

- **V1: Manually tuned configuration** The first complete configuration (V1) is determined manually using the procedure described above. The resulting weights of the different regularizations differ by orders of magnitude due to the varying scales of the loss terms and their respective sensitivities.
- **V2: Bayesian optimization** To further refine the setup, a Bayesian optimization is performed over the parameter space of regularization parameters defined in Table A.10. The best-performing trial, based on minimal surface loss on the test set, is selected as the V2 configuration. This version included a notable addition — the regularization on outputs — which is absent from V1.

Table A.11 compares the two configurations. The Bayesian search leads to substantially lower weights for the curvature and tilt penalties, suggesting that weaker geometric constraints improve performance. The dip margin is reduced by an order of magnitude, producing sharper enforcement of edge dips. The z-limit weight remains in the same range.

Additional Regularizations
In addition to the regularization terms included in the V1 and V2 configurations, two additional regularizations will be tested individually to assess their potential impact on training dynamics. These are the *regularization on outputs*—already integrated into the V2 setup—and the *repulsion regularization*, which aims to encourage diversity in the predicted surfaces. The corresponding parameters are summarized in Table A.12.

The regularization on outputs is tested individually, exploring the following decay strategies:

- Linear decay to zero at half of the total training epochs — strong influence early in training, reduced in later stages.
- Linear decay to zero at the final training epoch — maintains influence throughout the entire training process, but still gradually decreases its contribution.

The *repulsion regularization* includes two tunable parameters:

- Repulsion loss — minimum change in flux image loss required over a predefined number of epochs. If this condition is not met, the current surface is ”repulsed,” prompting exploration of alternative predictions. Repulsion threshold — loss value below which the repulsion mechanism becomes inactive.

The implementation and tuning of the various regularizations—ranging from geometric constraints such as curvature and edge dip penalties to exploratory mechanisms like repulsion—form the basis of further addressing the inherent ill-posedness of the inverse surface reconstruction problem. By systematically comparing the manually defined V1 configuration with the optimized V2 parameters from the Bayesian search, and by testing additional regularizations individually, their respective impacts on training dynamics and ill-posedness can be better understood.

With the individual components of the full approach now established, the next step is to illustrate how they work together within the full pipeline. The following section presents flow diagrams of the primary scripts, showing how dataset preparation, ray tracing, loss computation, and regularization are integrated into the complete training process. These diagrams progress from data preprocessing, to the main training loop, and finally to the incorporation of the regularization components, providing a step-by-step overview of the framework in operation.

4.2. Flow Diagrams of Primary Scripts

Having described the dataset design, ray tracing, loss functions, and regularization strategies in detail, it is illustrated next how these components interact in practice. The following section presents flow diagrams of the primary scripts. These diagrams first depict the data preprocessing pipeline, then the

main training loop, and finally the integration of the regularization components. Together, they offer a step-by-step representation of how the entire framework operates in practice.

4.2.1. Data Preprocessing

Before the neural network can learn from heliostat data, the raw measurements and simulated inputs must be transformed into a consistent, well-structured dataset. This preprocessing stage ensures that all training examples share the same coordinate conventions, remove any confounding effects of canting, and augment the data to improve model generalization. The process also prepares the training samples for efficient batch-wise raytracing, which will later feed the image loss computation.

In the raw PAINT database, the real surface measurements already include canting, and the ideal simulated surfaces also contain it. This means that canting and true surface deviations are combined in a single shape representation. To perform augmentation independently of position and canting, these components need to be separated. Furthermore, the original dataset is too small and too position-specific to effectively train a deep neural network without augmentation.

The preprocessing pipeline, illustrated in Figures 4.13 and 4.14, resolves these issues in two stages:

Stage 1 — Preparing canting-independent surfaces

As shown in Figure 4.13, the function generate_ideal_heliostats uses translation and canting vectors to produce ideal heliostat surfaces (with canting) for each position. The function get_z_displacement then loads both the ideal surfaces and their real counterparts from the PAINT database, subtracting the two to remove the canting component. The result is a tensor describing only the deviations from the ideal shape, independent of both canting and heliostat position.

These canting-independent surfaces are then split into training and test sets, followed by augmentation with generate_augmented_surfaces through rotations and interpolations. Augmentation is applied only to the training set, while the test set remains unaltered to stay consistent with Lewen et al [24].

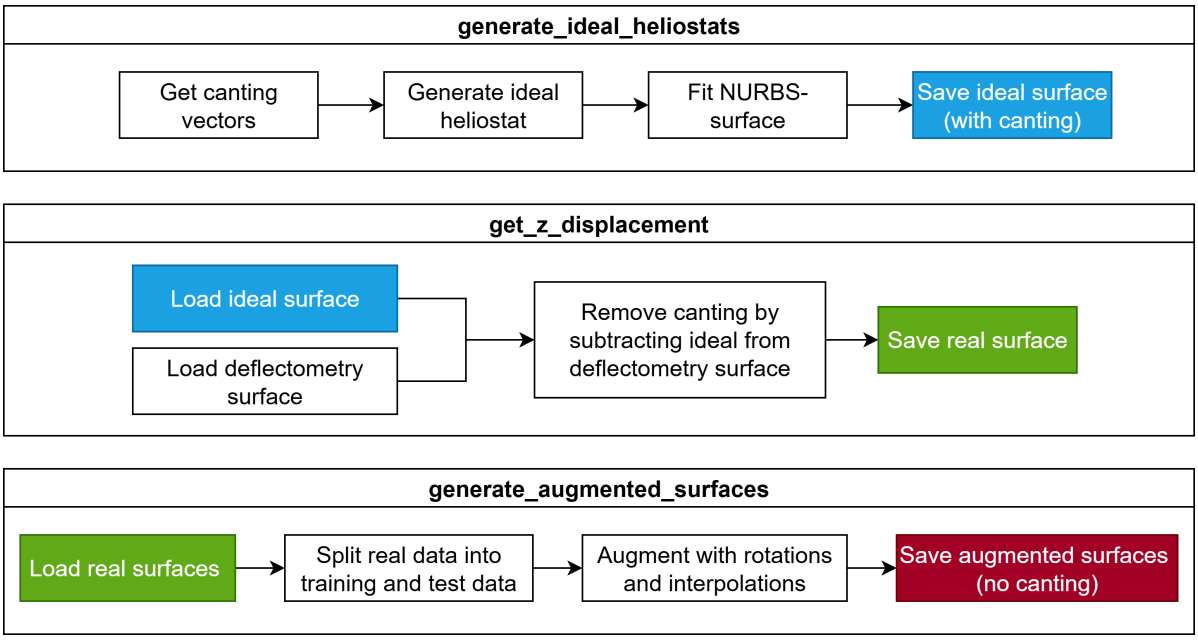


Figure 4.13: Workflow diagrams for generating ideal heliostats, extracting real surface displacements, and creating augmented training surfaces used in the dataset preparation.

Stage 2 — Generating flux density images

The second stage, shown in Figure 4.14, creates the optical output images needed for training. Each augmented surface is paired with a randomly selected heliostat position, and the corresponding ideal surface with canting is stored at the same index. These matched pairs form the simulated training

heliostats: real positions with real canting vectors from the ideal heliostat, combined with augmented but physically plausible surfaces.

For parallelization efficiency (Section 4.1.4), these matched heliostats are grouped into chunks of eight. For each chunk, the function overwrite_scenario_xyz_surface merges the ideal and augmented surfaces into a realistic surface with canting applied. It then calculates the surface points and normals required by the raytracer and writes them into the heliostat field parameters of the scenario.

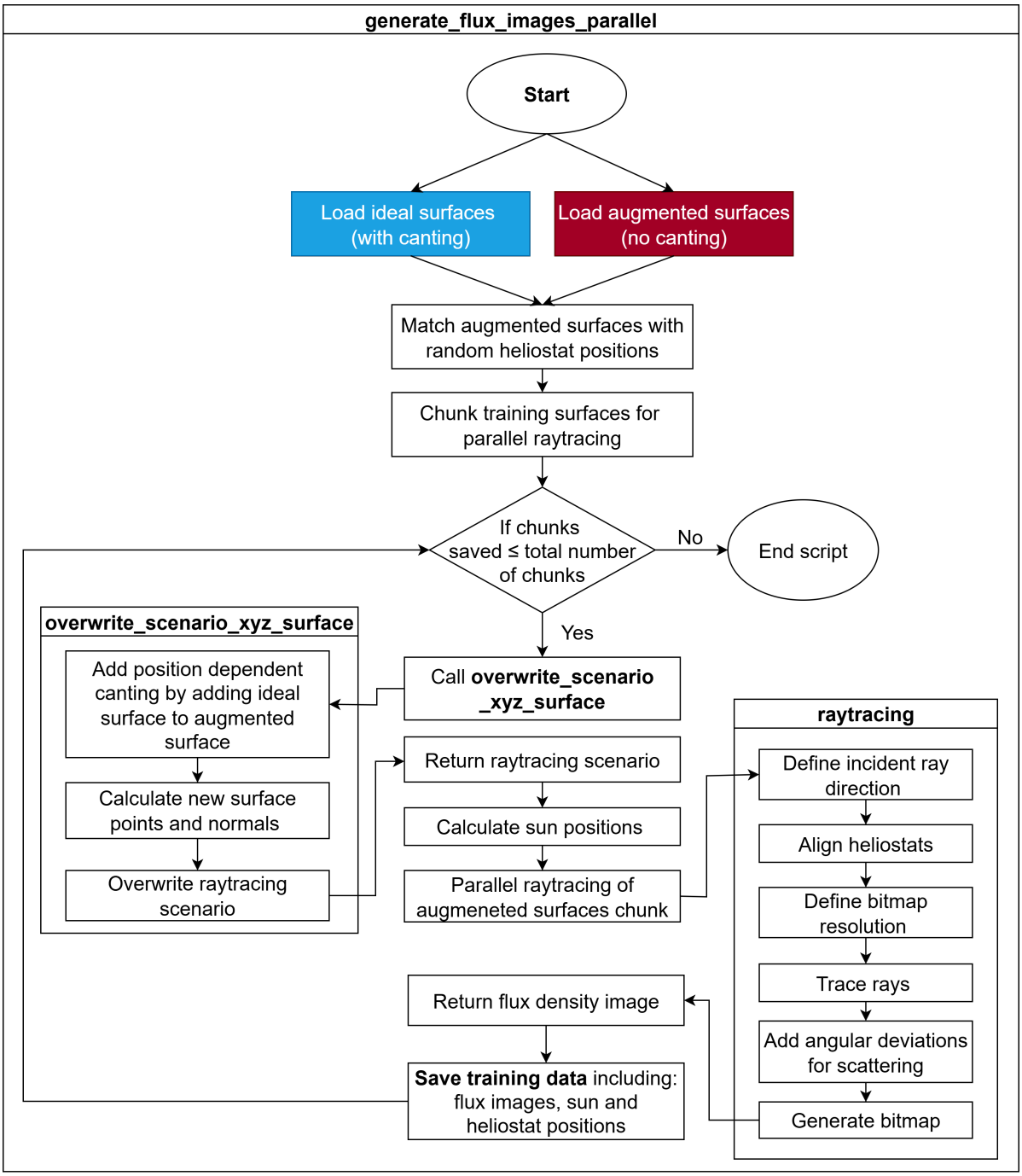


Figure 4.14: Workflow diagram of the `generate_flux_images_parallel` function, showing the main steps from loading surfaces to generating and saving flux density images for training.

Random but realistic Sun positions for the Jülich CSP plant are generated, and raytracing is performed in parallel for each chunk. The raytracing consists of the steps as described in detail in Section A.4.

The resulting flux density images are stored in the training dataset along with the Sun positions and heliostat positions used.

With preprocessing complete and flux density images generated, the framework is ready to integrate these datasets into the neural network training loop, described in the following section.

4.2.2. Training Loop

The training loop integrates all components introduced in the previous sections — from data preprocessing and neural network prediction to raytracing and loss computation — into a single, automated pipeline. Its purpose is to iteratively update the network weights so that predicted heliostat surfaces yield flux density images that match the measured references. The overall process is implemented in two main stages: the configuration of the run file, which defines all parameters and resources for training, and the execution of the train_dnn function, which performs the training iterations (Figures 4.15 and 4.16).

Stage 1: Configuring the run file

The run file serves as the entry point for the training pipeline, specifying all required inputs and hyperparameters. It first loads the raytracing scenario, allowing parameters such as the number of rays or aim points to be configured. While heliostat field parameters can be set arbitrarily at this stage, they will later be overwritten with the model predictions.

Both ideal surface tensors (including position-dependent canting) and real surface tensors from deflectometry are loaded. The ideal surfaces are necessary for integrating canting into the predicted geometry, while the real surfaces are used only for validation checks to measure how closely predictions match the ground truth. To avoid excessive memory usage, all large datasets are lazy loaded, meaning that only the required portions are held in memory at any given time.

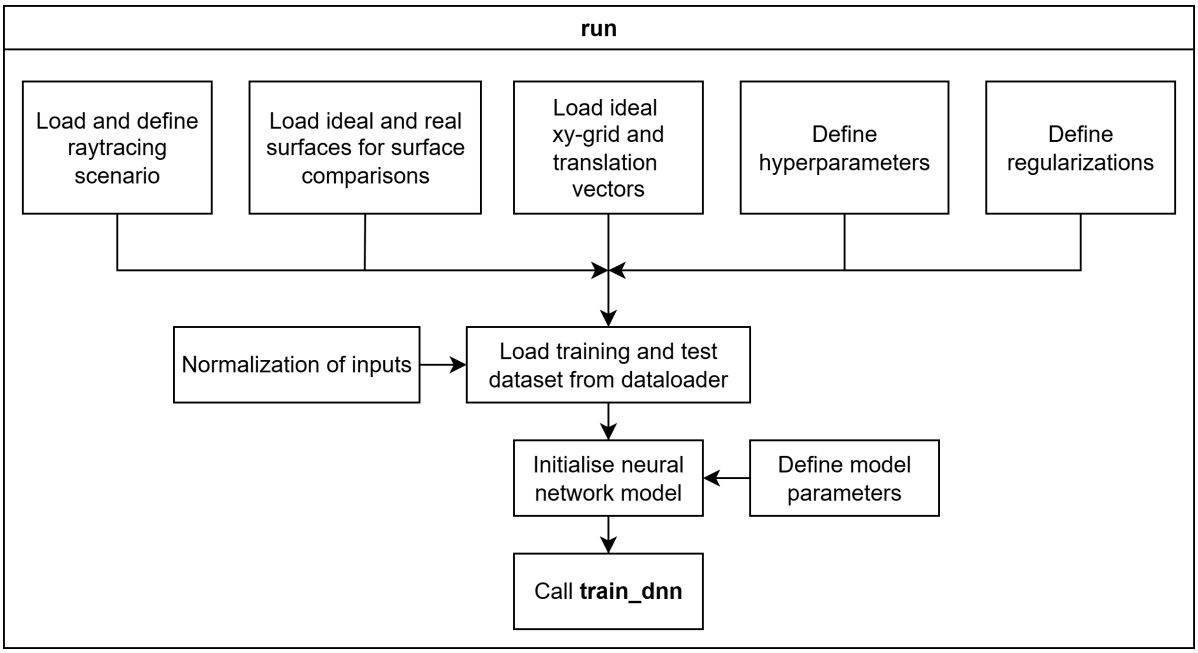


Figure 4.15: Workflow diagram of the `run` script, outlining the main steps of dataset preparation and model initialization.

The run file also loads:

1. The fixed x–y grid for control points
2. Translation vectors for each facet and heliostat
3. Hyperparameters such as learning rate, scheduler, weight decay, and number of training samples
4. The set of regularizations to apply during training (Section 4.1.9)

After normalizing all inputs to the range [0,1], the neural network model is defined and initialized. All parameters are then passed to the train_dnn function.

Stage 2: Executing the train_dnn function

The train_dnn function begins by setting up experiment logging with a TensorBoard writer. The training data are sampled and grouped into chunks of eight heliostats — a requirement imposed by the raytracer's parallelization strategy (Section 4.1.4) to ensure that all heliostats in a batch share the same Sun position.

For each batch, the neural network predicts the z-coordinates of the control points, outputting a 4×8×8 tensor per heliostat. These predictions are mean-centered to remove any vertical offset, ensuring comparability between surfaces. If regularizations are enabled, their penalties are computed here. The predictions are then passed to overwrite_scenario, which reconstructs the full NURBS surface by combining the fixed x–y grid with the predicted z-coordinates, applying facet translation vectors, and adding the ideal surface to restore canting. The corresponding surface points and normals are computed and written into the heliostat field parameters.

The updated scenario is then used for raytracing in parallel, producing a normalized flux density bitmap for each heliostat. The image loss between the predicted and reference bitmaps is computed, and regularization penalties are added if applicable. This total loss is backpropagated through the network, updating the model weights.

At the end of each epoch, the training loss and the surface loss is compared to the best recorded value. If improved, the model parameters are saved. An early-stopping mechanism halts training if the loss plateaus, preventing unnecessary computation.

Trainging Loop Summary

In summary, the training loop combines:

- Pre-processed, canting-corrected datasets
- Neural network predictions of heliostat surfaces
- Integration of predictions into a raytracing scenario
- Parallel computation of flux density images
- Loss calculation combining image and regularization terms
- Automated checkpointing and early stopping

Together, these steps form a structured and repeatable process for addressing the inverse heliostat surface reconstruction problem.

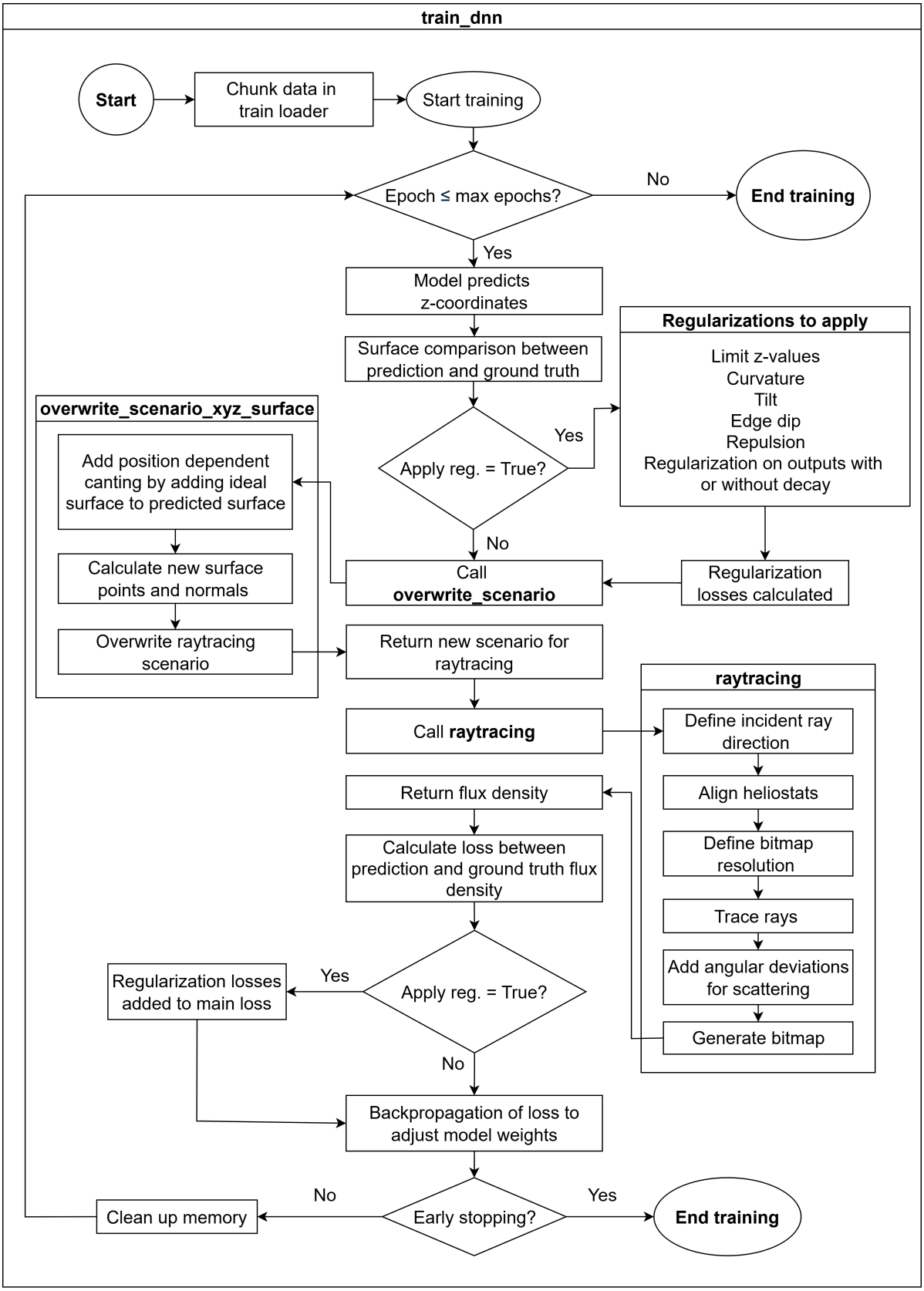


Figure 4.16: Flow diagram of the `train_dnn` function, illustrating the training process of the neural network. The workflow includes surface prediction, raytracing, loss computation (including optional regularizations), and weight updates through backpropagation, with iterative checks for stopping criteria.

4.2.3. Software

The implementation of the proposed approach described in the flow diagrams is developed in Python 3.10.0, a widely used programming language for scientific computing and machine learning. In the context of integrating machine learning models with physics-based processes, an efficient framework is required to support both high-performance numerical computations and automatic differentiation. To address this requirement, the PyTorch framework is employed in combination with other standard Python libraries [44]. PyTorch utilizes tensors as its fundamental data structure, which can be formally regarded as generalized arrays or multidimensional matrices capable of storing variables, intermediate results, and gradients required for backpropagation during neural network training.

All relevant functions and scripts for the described approach can be found via GitHub [45].

4.3. Methodology Summary

In summary, the methodology integrates pre-processed and canting-corrected datasets, a structured training pipeline, and targeted regularization strategies to address the ill-posed nature of the inverse heliostat surface reconstruction problem. The flow diagrams above illustrate the interaction between these components—from data preparation and neural network surface prediction to raytracing and loss optimization—highlighting the coherent structure of the approach. This framework provides the foundation for the experimental investigations presented in the following chapter.

5

Results

This chapter presents the experimental findings of this work, evaluating the ability of the proposed physics-aware deep learning framework to reconstruct heliostat surfaces from flux density images. The results are structured to assess the general training behavior and analyze the impact of regularization strategies. Finally, the nest performing models are compared against the supervised state-of-the-art.

5.1. General Results

This section presents the outcomes of the full training pipeline. The primary focus is on evaluating the training loop itself and the role of regularization strategies in enabling the network to reconstruct heliostat surfaces from flux density images.

Preliminary experiments confirmed that, without constraints, the network fails to converge to meaningful flux density predictions. For this reason, even configurations not labeled as implementing regularizations include a z-value limit to prevent surfaces from drifting outside physically plausible bounds (Section 4.1.9). In the following, the term *regularizations* therefore refers to additional constraints beyond this baseline. It will also sometimes be abbreviated to *regs* in naming conventions.

Model performance is assessed using two complementary metrics introduced in Section 4.1.6:

- The MAE between predicted and ground truth flux density images, which serves as the primary indicator of training quality. Accurate image reconstruction is treated as a prerequisite for surface prediction, since the flux density distribution is the observable input in the inverse problem.
- The MAE between predicted and real heliostat surfaces, which provides a direct measure of reconstruction accuracy once image alignment is achieved.

During training, both metrics are logged at regular intervals, and qualitative surface visualizations are generated every ten epochs. Final results are reported on the test set.

To systematically evaluate the influence of different design choices, a structured series of experiments was conducted. Starting from the baseline configuration obtained by grid search, individual factors—such as regularization strategies, data parameters, and network settings—were varied one at a time. The outcomes are summarized in Table 5.1, which lists for each experiment the number of training samples, the resulting flux image MAE, the surface MAE, and a short description of the condition applied.

This experimental design isolates the effect of each component on model performance and establishes a foundation for the discussion of trade-offs and implications in Chapter 6.

Table 5.1: Summary of experimental results across different configurations. The table lists the number of training samples, flux MAE, and surface MAE for each experiment group. Best-performing values are highlighted in bold.

Group	Experiment	Training Samples	Flux MAE	Surface MAE [m]
Baseline	Standard	5 000	2.09×10^{-2}	3.14×10^{-4}
	Surface model checkpoint	50 000	2.08×10^{-2}	3.06×10^{-4}
	Train model checkpoint	50 000	1.77×10^{-2}	3.83×10^{-4}
Loss Function	MAE (standard)	5 000	2.09×10^{-2}	3.14×10^{-4}
	SSIM loss	5 000	2.06×10^{-2}	3.81×10^{-4}
Training Sample Scaling	5k training samples (standard)	5 000	2.09×10^{-2}	3.14×10^{-4}
	10k training samples	10 000	2.34×10^{-2}	3.18×10^{-4}
	50k training samples	50 000	2.08×10^{-2}	3.06×10^{-4}
Hyperparams	Grid search (standard)	5 000	2.09×10^{-2}	3.14×10^{-4}
	Bayesian opt.	5 000	2.88×10^{-2}	3.12×10^{-4}
Regularizations	All regs V1 (manual)	5 000	2.89×10^{-2}	2.98×10^{-4}
	All regs V1 (manual)	50 000	2.76×10^{-2}	2.89×10^{-4}
	All regs V2 (Bayes)	5 000	5.19×10^{-2}	2.55×10^{-4}
	Repulsion	200	3.30×10^{-2}	3.21×10^{-4}
	Repulsion	5 000	2.01×10^{-2}	3.16×10^{-4}
	Outputs reg lin. decay, max epochs	5 000	2.73×10^{-2}	2.88×10^{-4}
	Outputs reg lin. decay, half epochs, surface model	5 000	$3,78 \times 10^{-2}$	2.55×10^{-4}
	Outputs reg lin. decay, half epochs, train model	5 000	$\mathbf{1.55 \times 10^{-2}}$	3.49×10^{-4}
Synthetic Datasets	Close 30–70m	5 000	4.92×10^{-2}	2.85×10^{-4}
	Close 30–70m, canting as input	5 000	6.94×10^{-2}	6.61×10^{-4}
	Close 5–15m	5 000	4.42×10^{-2}	$\mathbf{2.47 \times 10^{-4}}$

The individual experiments and their objectives are outlined in the following, following the order of Table 5.1.

5.1.1. Baseline

The baseline experiments establish the reference performance against which all other configurations are compared. Using 5 000 training samples, the standard configuration achieved a flux MAE of 2.09×10^{-2} and a surface MAE of 3.14×10^{-4}.

In addition, two model checkpoints were compared:

1. **Best surface model:** the checkpoint reached during training where the surface MAE is minimal. This model provides the most accurate surface reconstruction.
2. **Best training model:** the checkpoint with the lowest overall training loss, which also corresponds to the lowest flux density loss. This model typically appears near the end of training.

The distinction between these two checkpoints is illustrated in Figure 5.1, where the surface MAE reaches its minimum earlier, while the flux MAE continues to decrease throughout training. This effect is also visible in Table 5.1, where the surface-optimal checkpoint yields the lowest surface MAE in the baseline group, while the training-optimal checkpoint achieves a lower flux MAE. As the training

loop primarily minimizes the flux density loss, it continues optimizing flux prediction beyond the point where the surface prediction is most accurate. This behavior reflects the ill-posed nature of the inverse problem.

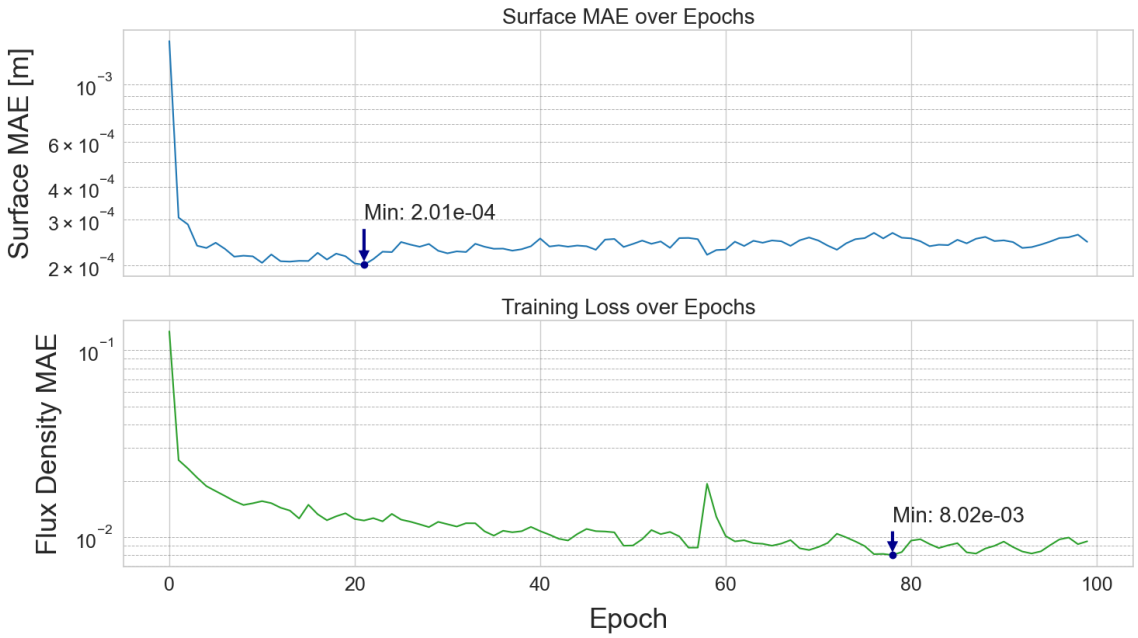


Figure 5.1: Training curves of the surface reconstruction error (Surface MAE) and the flux density error (Flux Density MAE) over epochs. The surface MAE reaches its minimum at an earlier stage of training, while the flux density MAE continues to decrease for a longer duration. Markers indicate the minimum values for each metric, corresponding to the surface-optimal and training-optimal checkpoints, respectively. As these values occur during training, they do not hold weight yet until the model is tested on the test set.

Unless stated otherwise, the surface-optimal model is evaluated in the following experiments.

5.1.2. Loss Function

One additional experiment using the SSIM loss function within the training loop was conducted to validate the choice of MAE as the primary error metric. The SSIM-based experiment resulted in a flux MAE of 2.06×10^{-2} and a surface MAE of 3.81×10^{-4}. This confirmed MAE as the more suitable choice for subsequent experiments, as Flux MAE is comparable while surface MAE performs worse with SSIM. The discrepancy to the results found in Table A.4 may be explained by the now much higher number of training samples.

5.1.3. Training Sample Scaling

The effect of dataset size was evaluated by training with 10 000 and 50 000 training samples, in addition to the standard 5 000. With 50 000 training samples, the model achieved the lowest overall error within the training sample scaling group (2.08×10^{-2} flux MAE, 3.06×10^{-4} surface MAE). However, the improvements over smaller datasets were marginal, and no consistent trend was observed across different numbers of training samples. Increasing the dataset size beyond 5 000 training samples yields only limited improvements, indicating diminishing returns from additional training samples and longer runtime.

5.1.4. Hyperparameters

The baseline hyperparameter configuration was derived from a grid search. To test whether a more automated tuning approach could further improve performance, Bayesian hyperparameter optimization was applied. It obtained a configuration with a cyclic learning rate. However, it achieved slightly worse flux prediction (2.88×10^{-2} flux MAE) while maintaining a surface MAE of 3.12×10^{-4}, showing no clear advantage over the baseline.

5.1.5. Regularizations

Several regularization strategies were evaluated, both individually and in combination. The V1 configuration resulted in a slightly lower surface MAE compared to the baseline, but at the cost of a higher flux MAE. Increasing the number of training samples only marginally improved both error metrics. In contrast, the V2 configuration produced a substantial reduction in surface MAE, but the flux MAE increased by a much higher margin. The predictions on the test set can be found in Appendix A.9.5.

Among the individually tested regularizations, the most effective was the output regularization with linear decay, where the weight decayed to zero by half the training epochs and the checkpoint was selected based on the best training loss. This setup achieved the lowest flux MAE overall (1.55×10^{-2}), though accompanied by a higher surface MAE (3.49×10^{-4}). With the exception of repulsion, all regularization strategies reduced surface MAE compared to the baseline, but consistently increased flux MAE, underscoring a trade-off between flux accuracy and surface reconstruction.

5.1.6. Synthetic Datasets

Finally, synthetic data test cases were conducted to examine whether reducing the heliostat–receiver distance could mitigate the ill-posedness of the inverse problem. Two distance ranges were evaluated: 30–70 m and 5–15 m. Both closer range yielded lower surface MAE values compared to the baseline. An additional experiment included canting values as inputs, but this did not improve performance compared to the non-canting case. The 5–15 m configuration achieved the best overall surface MAE (2.47×10^{-4} at 5 000 training samples), confirming that shorter heliostat–receiver distances improve surface reconstruction accuracy.

5.1.7. Summary

In summary, the experiments demonstrate that the standard configuration provides a consistent reference performance, while neither alternative loss functions nor hyperparameter optimization led to substantial improvements. Increasing the number of training samples had limited effect, suggesting diminishing returns beyond 5 000–10 000 samples. Regularization strategies highlighted a clear trade-off between flux and surface accuracy, with output regularization producing the best flux results but worse surface predictions. The synthetic distance experiments showed that shorter heliostat–receiver distances improve surface reconstruction, and lead to the best surface MAE, supporting the hypothesis that stronger optical signals aid the inverse mapping.

Building on these findings, the following sections focus on the two configurations that achieved the best results: the model with the lowest flux density image loss, and the model with the lowest surface MAE.

5.2. Lowest Flux Density Loss

The best result in terms of flux density image prediction is obtained by saving the best training model from a configuration using 5 000 training samples, where output regularization employed a linear decay reaching zero at half the number of epochs. This setup achieves an average flux density image MAE of 1.55×10^{-2}, the lowest among all experiments.

Figure 5.2 illustrates the behavior of this model. Part (a) shows the training progression of both flux density image and surface predictions. The flux density image converges rapidly, with only minor improvements beyond epoch 10, while the surface prediction initially approaches the correct shape but subsequently stagnates at an inaccurate geometry. Part (b) presents predictions for three randomly selected flux images from the test set, representing the best flux image prediction results obtained in this study.

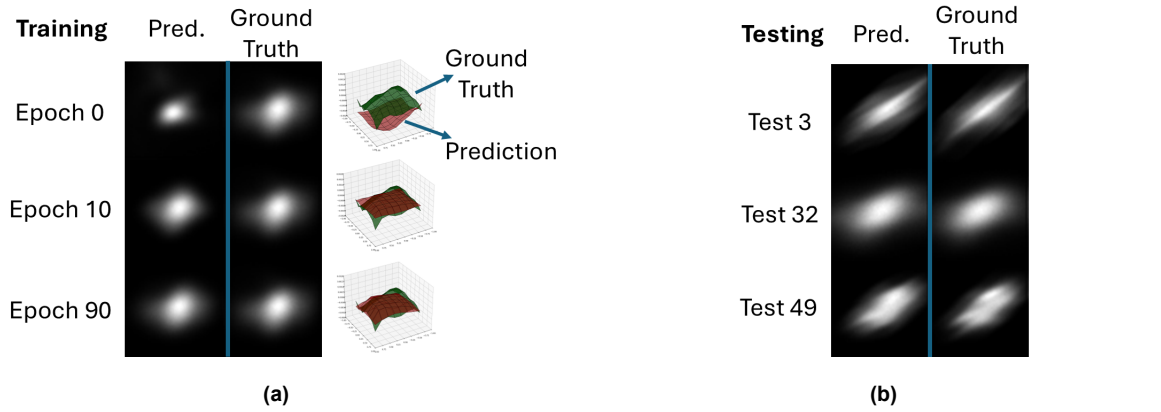


Figure 5.2: Flux density image predictions obtained with the best-performing flux density model. (a) Training progression at selected epochs, showing flux density predictions alongside corresponding surface reconstructions and ground truth. (b) Test predictions on three randomly selected surfaces, demonstrating the model's ability to reproduce flux density images from unseen data.

Figure 5.3 compares the predicted surfaces with the ground truth surfaces from the test set. Although the model successfully captures certain surface features—such as the tendency for edge control points to exhibit lower z-values and center points to remain more positive—it does not accurately reproduce the overall surface geometry. This results in a relatively high surface MAE of 3.49×10^{-4}, despite the good alignment of the corresponding flux density images.

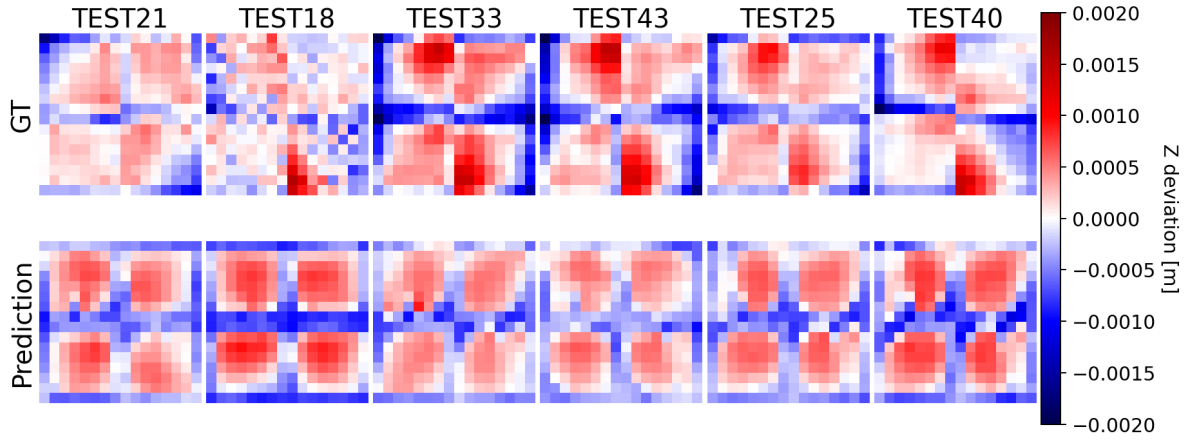


Figure 5.3: Surface prediction test results of the best flux density prediction model for six randomly selected test surfaces. Ground truth (GT) and predicted surfaces are shown for comparison, with color indicating deviations in the z-direction.

The resulting flux density accuracy statistics, computed as described in Section 2.3.1, are summarized in Table 5.2. For comparison, Lewen et al. [24] reported a median flux prediction accuracy of 92% using their supervised surface estimation approach, whereas the pipeline developed in this work achieves 84%. This outcome is unexpected, given that the present training loop explicitly optimizes flux image loss, while Lewen et al. target surface prediction as their primary objective. Overall, the results indicate that the proposed unsupervised framework achieves flux density prediction performance that is clearly improved compared to the ideal assumption, yet remains consistently below the accuracy obtained with the supervised approach.

Table 5.2: Comparison of flux prediction accuracy statistics across methods. Results are reported for the supervised approach of Lewen et al. [24], their ideal assumption baseline, and the unsupervised pipeline developed in this work.

Accuracy Statistic	Lewen et al. [24]	Ideal Assumption [24]	Pipeline
Minimum	0.43	0.37	0.54
Q1 (25%)	0.90	0.59	0.80
Median	0.92	0.67	0.84
Q3 (75%)	0.94	0.74	0.89
Maximum	0.98	0.90	0.93

While this subsection highlighted the configuration that achieved the most accurate flux density predictions, the primary objective of this work remains the accurate reconstruction of the heliostat surface geometry. Therefore, the model that achieved the lowest surface MAE during training is examined in the following section.

5.3. Lowest Surface Loss

The lowest surface reconstruction error is achieved with a synthetic dataset in which the heliostats are positioned much closer to the receiver than in the standard configuration. This setup reduces flux density prediction performance on the test set but yields the best surface MAE across all experiments, reaching 2.47×10^{-4} m. Representative flux density predictions from this model are shown in Figure 5.4.

Part (a) illustrates the training progression. As described in Section 4.1.1, the synthetic dataset differs from the standard case because canting is not applied correctly for the new position. As a result, reflections do not focus perfectly on the receiver and produce broader flux patterns. To capture these patterns, the target area was enlarged accordingly. Under these conditions, the model learns the flux distribution quickly: the loss stabilizes within roughly 10 epochs, with only minor improvements thereafter. The example surface shown on the right converges toward the general shape of the ground truth but still exhibits geometric discrepancies.

Part (b) presents several flux density predictions from the test set. Their visibly lower quality compared to the best flux-image model in the previous section reflects the increased flux MAE, despite the improvement of surface prediction.

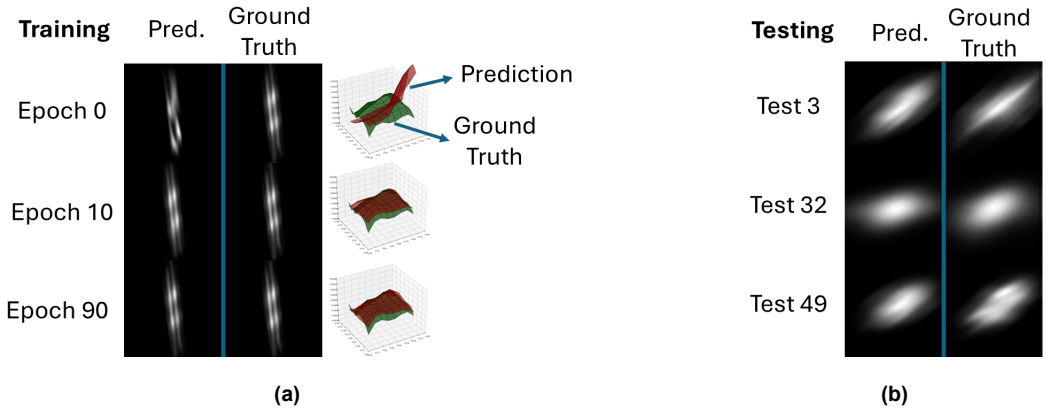


Figure 5.4: Flux density image predictions obtained with the best surface prediction model. (a) Training progression at selected epochs, showing flux density predictions alongside surface reconstructions and ground truth. (b) Test predictions on three randomly selected surfaces, illustrating lower flux prediction quality compared to the best flux-image model, despite improved surface prediction.

Nevertheless, the corresponding surface reconstructions show improvement, as illustrated in Figure 5.5. Although visible discrepancies remain between predicted and actual surfaces, especially in magnitude, this model captures the overall facet geometry slightly more faithfully than the flux-focused model in

Figure 5.3. In particular, the facet edges exhibit greater variation, rather than the uniformly negative displacement observed in earlier results.

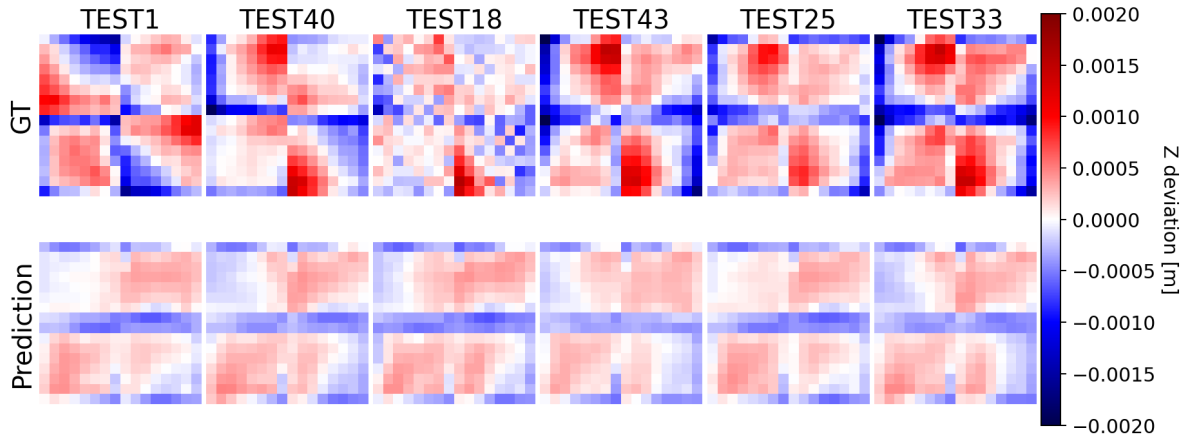


Figure 5.5: Surface prediction test results of the best surface prediction model for eight randomly selected test surfaces. Ground truth (GT) and predicted surfaces are shown with color-coded z-deviation. Compared to the flux-focused model, the predictions capture facet edges with greater variation, although discrepancies in magnitude remain.

In some cases, such as the first and third surface from the right, the general predicted pattern aligns with the general shape of the ground truth, though the model consistently underestimates the absolute z-values, as indicated by the lighter shading. This tendency is visible across all predictions. In contrast, other examples—such as the third surface from the left—show more pronounced deviations from the ground truth.

It should also be noted that not all ground truth surfaces exhibit a clearly interpretable pattern, as the TEST18 ground truth in Figure 5.5 shows. Part of this variability may arise from measurement noise or inaccuracies in the deflectometry data itself.

A comparison with the mean of the 5000 augmented training surfaces, shown in Figure 5.6 as the mean ground truth surface, indicates that the predictions capture several of the global characteristics of the data. Both the individual TEST43 prediction and the mean test surface prediction generally exhibit depressed facet edges and elevated centers, which are also present in the mean ground truth. However, in the predictions these edge regions are less sharply defined, and the boundaries between elevated and depressed areas appear smoother.

In the upper-left and lower-right facets, the central shape of the mean ground truth is reproduced reasonably well, albeit with a reduced magnitude in the predictions. By contrast, the top-right and bottom-left facet shapes are reconstructed less smoothly. Nevertheless, the predicted deviations are closer in order of magnitude to the ground truth than the other two facets.

Overall, the TEST43 prediction is highly similar to the mean test surface prediction, consistent with the behavior observed for other samples in Figure 5.5. This suggests that the network predominantly learns an underlying global mean pattern characteristic of the dataset, while showing limited ability to reconstruct fine-grained variations of individual heliostats.

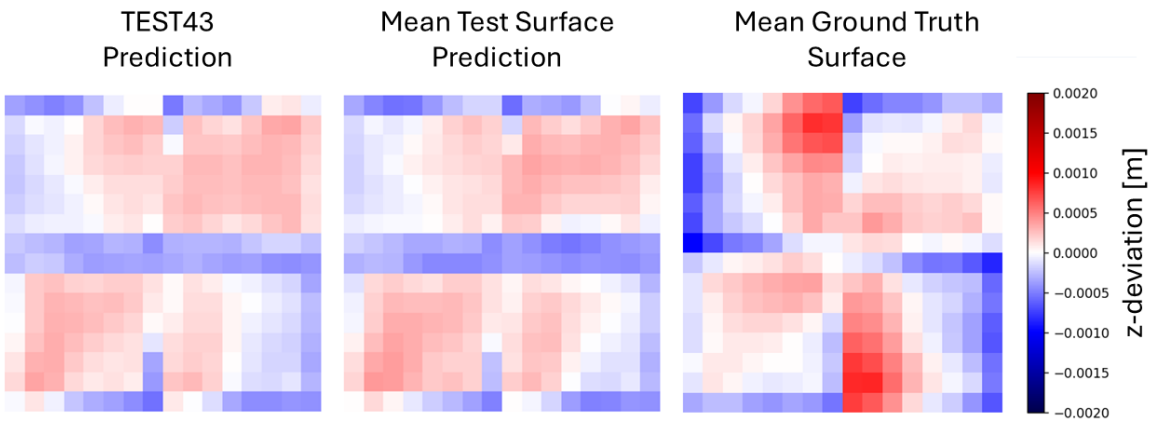


Figure 5.6: Comparison of predicted and ground truth mean surfaces. From left to right: prediction for a single test heliostat (TEST43), mean prediction across all test heliostats, and mean of the 5000 augmented training surfaces (ground truth). Color indicates facet-wise z-deviation in meters.

Table 5.3 presents a comparison of the surface MAE results on the test set, using sum normalization to ensure consistency with the evaluation in Lewen et al. The median surface MAE achieved by the best-performing model is 2.4×10^{-4} m, which is notably higher than the surface MAE of 1.4×10^{-4} m reported by Lewen et al. on the same test set of heliostats [24].

Table 5.3: Comparison of surface MAE statistics between the supervised approach of Lewen et al. [24] and the unsupervised pipeline developed in this work. Reported values include the minimum, first quartile (Q1), median, third quartile (Q3), and maximum MAE across the test set of heliostats.

Surface MAE Statistic	Lewen et al. [24] [m]	Pipeline [m]
Minimum	0.00007	0.00011
Q1 (25%)	0.00012	0.00019
Median	0.00014	0.00024
Q3 (75%)	0.00017	0.00029
Maximum	0.00070	0.00047

Another comparison can be made with the results of Table A.7.1, where the deviations of real heliostat surfaces from their ideal shapes were quantified. The mean absolute z-displacement was found to be 3.7×10^{-4} m in the deflectometry data. Assuming an ideal surface would therefore introduce an average error of this magnitude. By contrast, the proposed pipeline achieves a median surface prediction MAE of 2.4×10^{-4} m, indicating that it reconstructs the true surfaces more accurately than the ideal-surface assumption.

Taken together, this indicates that the unsupervised pipeline, although not yet matching the 1.4×10^{-4} m accuracy of the supervised method by Lewen et al., nonetheless improves upon the simplistic assumption of ideal surfaces. The results therefore highlight both the promise of integrating deep learning with a physics-based training loop and the need for further refinements in training strategies, model architecture, and regularization design to close the remaining performance gap.

With the experimental results now presented and analyzed, the following chapter places these findings in a broader context. It reflects on their implications for inverse surface prediction, and critically examines the strengths and limitations of the proposed approach.

6 Discussion

This chapter builds upon the results presented in Chapter 5 by providing a detailed interpretation and critical evaluation. First, the experimental findings are discussed in depth. Subsequently, general limitations of the proposed approach are addressed, including both methodological and practical challenges. The chapter concludes with recommendations for future research directions aimed at further improving surface reconstruction performance and the overall robustness of the pipeline.

6.1. Reflection on Results

The reflection on the results follows the order of the experiments as presented in Table 5.1.

6.1.1. Best Surface vs. Best Flux Model

The first set of experiments compares the effect of selecting the best surface prediction model during training versus the best flux density prediction model. This comparison is evaluated during two configurations: the baseline setup with 50 000 training samples, and a configuration within the regularizations setup with 5 000 training samples combined with output regularization. Although several parameters differ between these cases, a consistent trend emerges: selecting the best surface model yields slightly lower surface MAEs, while selecting the best flux model produces lower flux image MAEs.

A notable pattern is that the best surface prediction model often arises very early in training, as illustrated in Figure 5.1. The surface MAE curves typically show a sharp initial decline, followed by a mild increase and eventual plateau in the range of 2×10^{-4} to 3×10^{-4}. This plateau reflects convergence toward an alternative surface that reproduces the flux density distribution well but deviates from the ground truth geometry.

In some cases, the best surface model is reached within the first five epochs, or even earlier. An illustrative example is the *50k training samples* experiment from Table 5.1, where the lowest surface MAE occurs in Epoch 1, prior to convergence toward a different solution with higher surface loss. Figure A.6 shows the corresponding training curve, highlighting how the apparent best surface model may arise very early, before convergence. This supports the observation that selecting checkpoints based on surface loss can favor unstable early solutions. The final test performance, however, remains comparable to other runs, as reflected in Table 5.1. This suggests that surface-based checkpoint selection can capture favorable early solutions, but prolonged training—driven by flux optimization—gradually shifts the model away from them.

6.1.2. Number of Training Samples

The first set of experiments discussed in Chapter 5 examined the impact of increasing the number of training samples to the neural network. Contrary to the initial expectation, a higher number of training samples did not lead to significant improvements in either the flux density image MAE or the surface MAE. Instead, the performance remained relatively stable, with only minor fluctuations. Two explanations are possible: either the network already converges to a valid solution using just 5 000 training

samples, or the number of training samples remains insufficient to produce a significant improvement in surface MAE.

Notably, across the three runs—where the number of training samples was the only varying parameter—the network consistently converged to similar, incorrect surfaces during training, as illustrated in Figure 6.1. While there are minor variations among the predictions, the overall shape is nearly identical. This explains the comparable prediction quality across these runs: the network quickly converges to a suboptimal surface, even with a low number of training samples.

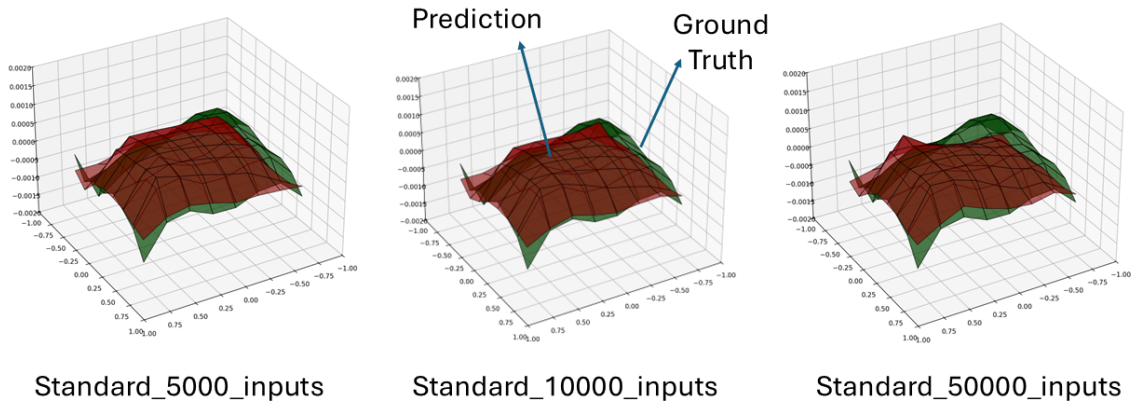


Figure 6.1: Predicted surface approximation of Facet 1 of heliostat AUG1 after training with 5 000, 10 000, and 50 000 samples. Across all three cases, the network converges to a similar but incorrect surface shape (red), differing from the ground truth (green). This indicates that increasing the number of training samples alone does not resolve the convergence to a suboptimal solution.

This behavior highlights the core limitation: the incorrect surface already yields a flux density image that is difficult to distinguish from the ground truth. Any deviation away from it risks degrading the flux prediction, even if it would move closer to the true surface. Addressing this issue requires additional regularization to penalize implausible surfaces and better constrain the inverse mapping.

6.1.3. Bayesian Hyperparameters

Most experiments used the standard hyperparameters obtained from the grid search (Table A.3). For comparison, one run applied the hyperparameters derived from Bayesian optimization, while keeping the loss function identical. This configuration did not yield improvements in either surface MAE or flux MAE, and was therefore not pursued further.

A possible explanation is that the Bayesian optimization was conducted under conditions that could be refined: the search space may not have been specified narrowly enough to capture effective regions, only 25 trials were run, and the evaluation was based on 200 training samples.

6.1.4. Regularizations

V1 & V2 Configurations

The Regularization experiment group in Table 5.1 presents experiments incorporating additional regularizations. The first configuration, V1, was derived through heuristic reasoning combined with local testing, as summarized in Table A.11. Each term was added directly to the loss, with weights typically contributing between 1% and 20%. The assumption was that a subtle influence would guide training without overwhelming the primary objective. In retrospect, this balanced weighting may have been too simplistic, since some terms may need to dominate more strongly to be effective.

The regularization-specific parameters were motivated by the deflectometry data of real heliostat surfaces. However, when applied jointly in V1, the regularizations did not greatly outperform the standard configuration: the flux density MAE worsened, while the surface MAE improved only marginally. Increasing the number of training samples only improved both outcomes slightly.

The resulting surface predictions of the V1 configurations are shown in Figure A.15 in the Appendix. The predictions appear overly smooth and flattened, with weak variation and a rigid, square-like structure. This suggests that some terms—such as the edge-dip regularization—imposed overly strong priors, leading to surfaces that matched the structural bias rather than the true geometry.

To address these shortcomings, a second set of regularization parameters, V2, was identified using Bayesian hyperparameter optimization, targeting the minimization of surface MAE.

The V2 configuration is presented in Table A.11. This setup marks a significant improvement in surface prediction quality, achieving the shared second-best surface MAE across all experiments, as shown in Table 5.1. However, the flux density image MAE remains relatively high, which is a recurring trade-off when surface accuracy improves.

Unlike previous runs, this was also the first experiment in which the surface MAE steadily decreased throughout training, as illustrated in Figure A.21. The error curve exhibits a consistent downward trend, without the early stagnation or reversal observed in other configurations. This suggests that the regularization weights found via Bayesian optimization allowed the network to avoid premature convergence toward an incorrect surface and instead continue refining its predictions across epochs. This behavior indicates that appropriately tuned regularization can alter the training dynamics in a favorable way, enabling the model to improve surface reconstruction beyond the early plateau observed in other experiments.

The most apparent differences to V1 is that the predicted surfaces during the testing are more rigid, less diverse, and even flatter, as illustrated by Figure A.19. This flatness leads to flux densities that emulate the shape of ideal surfaces, as shown by Figure A.20.

Taken together, the comparison between V1 and V2 highlights both the promise and the limitations of regularization. Bayesian optimization in V2 successfully improved surface MAE and stabilized the training dynamics, preventing the early stagnation observed in V1. However, this came at the cost of reduced geometric richness: the predicted surfaces were flatter and less diverse, and the corresponding flux densities resembled idealized shapes rather than realistic measurements. Thus, while V2 demonstrates that carefully tuned regularization can improve numerical accuracy in surface prediction, it also underscores that the current formulations remain too restrictive to fully capture the variability of real heliostat geometries. Further refinement of the regularization terms is therefore required to balance accuracy with realism.

Repulsion

The repulsion regularization was evaluated separately from the other terms, as it introduces a distinct concept. Instead of nudging predictions toward a specific geometry, it penalizes convergence to surfaces that no longer improve flux density prediction. Despite this novel formulation, the strategy did not improve either surface MAE or flux MAE relative to the standard configuration.

To give the mechanism sufficient opportunity to act, training was extended to 1 000 epochs. The training curve in Figure A.26 indeed showed repeated peaks and decays, consistent with transitions between surface solutions. The lowest surface MAE occurred only at epoch 433, and the corresponding model was saved and evaluated. However, test results still underperformed compared to the standard setup, suggesting that repulsion-induced transitions to new surfaces did not lead to genuinely better solutions.

Intermediate predictions (Figure 6.2) confirm that the network explored multiple surface states: the surface at epoch 304 is already close to the later minimum. While this demonstrates that the regularization encourages exploration, the final solutions failed to generalize to the test set. A likely cause is overfitting, as the number of training samples was reduced to 200 to keep runtime manageable under the extended schedule.

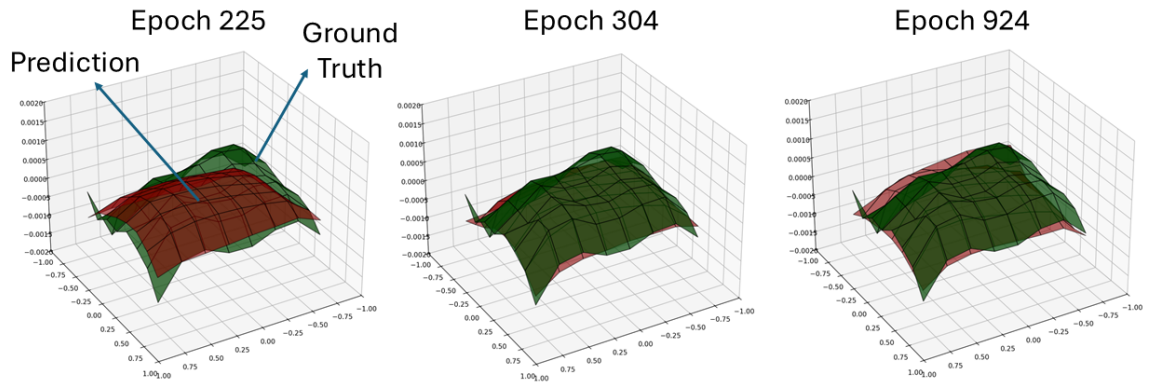


Figure 6.2: Predicted surfaces at different stages of training with 200 samples: epoch 225, epoch 304, and epoch 924. While the network explores multiple surface states, the predictions all converge toward a suboptimal shape (red) that deviates from the ground truth (green).

An additional run with 5 000 training samples and 500 epochs was carried out to test whether larger training sample sets might help the model discover more accurate solutions. However, this configuration also failed to outperform the baseline, with both flux and surface errors not showing significant improvement. This indicates that simply extending training or scaling training samples is not sufficient for the repulsion strategy to succeed.

Two main limitations emerged from this approach. First, the solution space may be too vast for repulsion alone to guide the model reliably toward the correct surface. Second, the mechanism may discard promising solutions prematurely, as its parameters were not systematically tuned. More fundamentally, even if the correct surface were reached, the model has no way of recognizing it without external surface information—contradicting the unsupervised objective of the pipeline. For these reasons, the repulsion regularization remains uncompetitive in its current form and was not pursued further.

Regularization on Outputs

The best flux prediction results were obtained by applying output regularization with a linear decay schedule, where the regularization weight was reduced to zero at half the maximum number of training epochs. As reported in Section 5.2, this approach achieved the lowest average flux MAE on the test set at 1.55×10^{-2}.

Figure 5.2 showed that the predicted flux images closely match the ground truth in both shape and intensity, with only minor—but still visible—deviations. Compared to the supervised approach of Lewen et al., this model achieves slightly lower accuracy, but it still represents a substantial improvement over the ideal-surface assumption. This highlights that realistic flux density predictions can be achieved within the unsupervised framework.

Importantly, the ability to generate surfaces that result in realistic flux density images is valuable in itself. Even if the surfaces are not perfectly accurate, producing physically plausible flux distributions contributes meaningfully to the broader goal of optimizing CSP plant efficiency. This method thus represents a step toward further automating flux density optimization in solar field calibration.

Figure 5.3 presented the corresponding surface predictions. The linear decay appears beneficial compared to a slower decay to zero, as it prevents over-regularization: the network is initially guided toward flatness but later given freedom to adapt to surface geometry. However, the predicted surfaces remain overly rigid, following the square-like structure observed in other experiments and deviating from realistic geometries. Interestingly, this structured pattern emerges even without applying the V1 or V2 regularizations, suggesting that the network has an intrinsic tendency to favor such shapes for reproducing flux images.

Taken together, these findings indicate that output regularization with linear decay is effective for improving flux accuracy but insufficient for accurate surface reconstruction. While this strategy strengthens the optical side of the inverse problem, additional constraints are required to counteract the network's

bias toward rigid, non-real surfaces. Future approaches may benefit from combining output regularization with more geometry-aware terms, such as those in the V1 configuration, to achieve a better balance between flux prediction and surface accuracy.

6.1.5. Synthetic Datasets

To address the ill-posedness of the inverse problem and discourage convergence toward incorrect surface predictions, a set of experiments investigated the effect of positioning the heliostats significantly closer to the receiver. The first attempt employed a range of 30–70 m, which yielded a slight improvement in surface MAE relative to the standard configuration. However, this gain came at the cost of a substantial increase in flux density image MAE.

This outcome is expected: at shorter distances, flux images contain finer details and distinct facet structures, making them inherently more difficult to predict. The redefined heliostat positions, combined with the lack of canting adjustments, further altered the flux characteristics, leading to sharper, less homogeneous patterns. While these changes provided stronger geometric cues for surface reconstruction, they simultaneously increased the complexity of flux prediction and reduced the model's ability to generalize across test cases.

To compensate for the altered flux patterns, an additional experiment introduced the canting vector as an extra input, aiming to provide the model with more contextual information. However, this modification had little effect: the predicted flux images were comparable—though slightly worse—than those of the experiment without canting, and the surface reconstructions degraded further, becoming almost structureless and yielding a substantially higher surface MAE. These outcomes suggest that, at least in the present setup, simply supplying canting information does not enhance reconstruction performance. Consequently, this approach was not pursued further.

A final experiment further reduced the heliostat distance to 5–15 m. The rationale was that positioning heliostats at very short ranges would yield flux density images with stronger spatial detail, thereby providing the network with richer information for surface reconstruction. This expectation was confirmed: the experiment achieved the lowest surface MAE of all runs, with a value of 2.47×10^{-4} m.

The results in Figures 5.5 and 5.6 demonstrate that the network has learned some structural patterns present across the dataset, such as depressed edges and elevated centers. However, the predictions are not yet competitive when compared to the supervised benchmark. While individual predictions can resemble the mean ground truth surface, this similarity is only partial: some facet regions and magnitudes still deviate noticeably from the mean ground truth. The strong similarity between individual predictions and the mean surface predictions also highlights a limitation: the model tends to regress toward an "average heliostat" and fails to reconstruct fine-scale, heliostat-specific deviations. This indicates that while the network is effective at capturing some dataset-level biases, it does not generalize sufficiently to the variability of individual geometries. The consistent underestimation of absolute z-values further suggests a conservative bias, where predictions remain within a reduced dynamic range.

Overall, the experiments with closer heliostat positions indicate that reducing the distance to the receiver can provide stronger geometric cues that improve surface reconstruction, albeit at the cost of flux density prediction accuracy. The results from the 5–15 m configuration suggest that such data may be useful as a pretraining step, allowing the network to internalize structural patterns before being fine-tuned on more realistic settings. However, the strong resemblance of the predictions to the mean training surface also highlights the risk of overfitting to dataset-level trends rather than faithfully reconstructing individual geometries. Future research should therefore investigate whether short-distance pretraining can be integrated without compromising generalization to realistic heliostat fields.

6.2. General Limitations

The previous section presented and reflected on the results obtained in Chapter 5. This section outlines general limitations that apply to the overall training procedure and experimental design. These limitations may have influenced the outcomes and provide important context for interpreting the results and guiding future work.

6.2.1. Dataset

This deep learning approach, while incorporating the physical characteristics of the system, faces several limitations. A primary concern is that all training data are fully simulated. Although efforts were made to enhance realism—such as basing the surface geometry on real deflectometry data and generating interpolated, augmented surfaces—these still do not fully represent actual heliostat surfaces found in Jülich or other CSP plants.

To mitigate this, 20% of the dataset was preserved as unaltered real surfaces to introduce a stronger bias toward physical realism. Nevertheless, even these real surfaces were paired with randomly assigned heliostat positions. In most experiments, these positions were drawn from real locations in the Jülich field, but in those investigating proximity to the receiver, entirely artificial and unrealistic positions were used.

As a result, the dataset does not reflect the true distribution of surface–flux relationships encountered in practice. This limitation implies that, even if the model performs well within the simulated framework, its ability to generalize to real-world heliostats remains uncertain and will ultimately need to be validated on measured flux data.

6.2.2. Augmentation

Another limitation lies in the way real surface data were augmented to generate the training dataset. Specifically, 20% of the training surfaces were kept as unmodified real surfaces—423 unique geometries placed at different heliostat positions. However, the training data were not shuffled, meaning that the first 10 000 surfaces consisted entirely of these repeated real examples. While the heliostat positions varied, the underlying surface geometry did not, limiting the diversity of the training samples in these smaller datasets.

As a result, all experiments with 10 000 surfaces or fewer were effectively trained only on variations of the same 423 geometries. This restricted diversity may have encouraged overfitting and reduced generalization, since the model was not exposed to the broader set of augmented geometries. In contrast, experiments with 50 000 surfaces included a larger proportion of augmented data, which should, in theory, support stronger generalization. However, this benefit was not clearly reflected in the results.

One possible explanation is that the augmentations themselves did not introduce enough variability to meaningfully expand the training distribution. This interpretation is consistent with Figure 5.6, where the mean ground truth surfaces of the 5 000- and 50 000-surface datasets appear very similar. Although augmentation techniques such as surface rotation were applied, their overall effect on the geometry distribution may have been too limited to significantly influence model performance.

Taken together, these findings suggest that the augmentation strategy did not sufficiently enrich the training set, limiting the model's ability to escape biases toward "average" surface patterns. More sophisticated augmentation methods may therefore be required to improve robustness.

6.2.3. Canting

Another limitation arises from the way canting is handled in the generation and augmentation of surface data. In the current approach, canting values are separated from the real surfaces and later re-applied to ideal heliostats that incorporate these canting vectors. This process assumes that canting can be represented as a simple translational offset in the z-direction for each control point. However, in reality, canting involves not only vertical displacement but also a slight angular rotation of the surface.

This rotational component is currently neglected. As a result, when subtracting the ideal heliostat shape from the real, canted heliostat to isolate the surface deformation, the z-direction correction ensures that control points are locally aligned. However, due to the omission of rotation, the interpolated NURBS surface between control points does not accurately reflect the true surface shape. This discrepancy is then propagated when the surface is added to a new ideal heliostat at a different position during the augmentation process.

Importantly, these rotational errors do not cancel out. Since each heliostat position is associated with different canting properties and thus a different angular offset, small but systematic inaccuracies are

introduced during surface augmentation. These errors, though minor, create an inconsistency between the intended and actual surfaces used in training.

Within the simulated environment, this limitation does not undermine the validity of the results, since all surfaces are generated and treated consistently. However, before applying the model to real measurement data, the canting procedure must be revised to incorporate both translational and rotational effects more accurately. Otherwise, the mismatch between simulated and physical canting would risk introducing systematic errors into the reconstruction process.

6.2.4. Parallel Raytracing

Loss Evaluation Restricted to One Sun Position

Another key limitation arises from how the raytracing component is integrated within the training loop, particularly in the calculation of loss (see Section 4.1.4). In the current implementation, each predicted surface is evaluated by raytracing it under only a single Sun position—the first of the eight available for that sample. The resulting flux density image is then compared to its ground truth counterpart, and the loss is computed solely on this image. Although the network training samples contains eight flux images (one per Sun position) together with the heliostat position, only one of these views contributes to the feedback signal during backpropagation.

This design was chosen for efficiency: raytracing is performed batch-wise across multiple predicted surfaces for the same Sun position during training data generation, which greatly reduces runtime. However, the current implementation does not allow a single predicted surface to be raytraced under all eight Sun positions within the loop. As a result, seven potentially informative flux images are discarded, and the loss reflects only one view of the surface

Including all eight Sun positions in the loss computation would provide a richer and more reliable learning signal. Each predicted surface would then be tested under multiple lighting conditions, reducing ambiguity in the inverse problem and encouraging convergence toward geometries that are consistent across different views.

By relying on only one Sun position, the model risks learning surfaces that reproduce flux patterns accurately for that specific view but perform poorly for others. This imbalance may partly explain why reconstructions sometimes converge to surfaces that are optically plausible (under one Sun position) but structurally inaccurate overall.

Looking ahead, extending the loss calculation to multiple Sun positions appears to be an important step toward improving robustness. Ideally, each predicted surface would be evaluated under all eight Sun positions, producing eight flux images and corresponding loss terms instead of just one. While this would increase computational cost—roughly by a factor of eight—the overhead may be manageable with improved parallelization strategies. Balancing efficiency with representational fidelity will therefore be an important consideration for future work.

Limited Sun Position Variability across Batches

Another limitation stems from how Sun positions are handled during training. To enable parallel raytracing as described in Section 4.1.4, all surfaces within a given batch are required to share the same set of eight Sun positions. The current implementation does not support evaluating each surface under its own randomly sampled Sun positions within the same batch.

This design simplifies processing but comes at the cost of reduced variability: instead of every surface being exposed to unique illumination conditions, entire batches rely on identical Sun positions. While experiments with 50,000 training samples still included 6,250 distinct sun–position sets—providing substantial diversity at the dataset level—the within–batch uniformity limits the model's exposure to fine-grained variability that could otherwise improve generalization.

The restriction also introduces practical drawbacks. Because only one set of Sun positions can be used per batch, batch sizes must remain relatively small to preserve diversity across training. Larger batch sizes would either require regenerating the full training dataset with new sun–position assignments or risk further reducing variability. This lack of flexibility not only increases runtime but also complicates systematic exploration of batch-size effects on training performance.

Looking ahead, enabling independent sun–position sampling per surface within a batch could simultaneously increase variability, improve runtime efficiency through larger batch sizes, and allow more flexible experimentation.

6.2.5. Computational Efficiency

A further limitation concerns the substantial computational cost of training. For example, training with 5 000 training samples over 100 epochs typically requires around 12 hours, while the same setup with 50 000 training samples can take 5–6 days. Much of this cost stems from the raytracing and loss evaluation components, which dominate runtime and highlight the need for further optimization.

In contrast, inference is highly efficient once a model has been trained. Both flux density images and surface predictions can be generated within minutes, making the approach practical for real-time or near–real-time deployment in calibration or monitoring scenarios.

Taken together, these observations emphasize an important trade-off: while the training procedure is computationally expensive, its outcomes can be used efficiently for inference in downstream applications. This makes the approach well-suited for deployment once a model is available. However, the high training cost limits the ability to explore design choices and iterate quickly during development. Future work should therefore also focus on accelerating training to make experimentation and refinement more feasible.

6.2.6. Ill-Posedness of the Inverse Problem

At the core of this work lies an inherently ill-posed inverse problem: predicting a continuous heliostat surface from limited and indirect observations, namely flux density images. From the outset, it was clear that this task is underdetermined, as many different surface shapes can generate visually similar flux distributions.

This ambiguity was repeatedly confirmed in the experiments and became most apparent in the repulsion-based training strategy. While the repulsion loss was intended to help the model escape suboptimal minima, its behavior instead highlighted the vastness of the solution space. The model frequently transitioned between distinct surface geometries that all produced comparable flux images, underscoring the lack of a unique solution under the current setup.

The findings indicate that, without stronger constraints or supervision—such as synthetic datasets with closer, though less realistic, heliostat positions—the model often converges to plausible yet structurally inaccurate surfaces. Nevertheless, the best-performing experiment demonstrated that certain dataset designs can enable the network to capture some structural patterns of the ground truth, suggesting that partial recovery of surface geometry is possible. Even so, these improvements remain limited in scope, and the underlying ill-posedness continues to restrict the model's ability to reconstruct surfaces faithfully across diverse cases. This constraint is not just a limitation of the present implementation but a property of the problem itself, and it will need to be addressed further in future research.

6.3. Recommendations and Future Research

This section outlines potential directions for improving the proposed method and guiding future research toward achieving more accurate heliostat surface predictions.

6.3.1. Data Augmentation

Future work could focus on introducing richer and more varied surface augmentations. Instead of relying on reusing a limited set of geometries across positions, new strategies could involve perturbations of facet curvature or localized deformations that mimic real manufacturing imperfections. Such diversity could help better approximate the range of surfaces encountered in practice and lead the model beyond its current tendency to predict average-like geometries.

6.3.2. Improved Parallel Raytracing

A key priority for future work is to extend the current loss mechanism in the training loop to incorporate all eight Sun positions. In the present setup, reconstructions often reproduce flux density patterns well but correspond to incorrect surface geometries, reflecting the ambiguity of the inverse problem. Multi-

Sun loss calculation could directly address this issue by testing each predicted surface under multiple illumination conditions, thereby narrowing the solution space and encouraging convergence toward geometries that remain structurally consistent. Similarly, removing the restriction of batch-wise Sun position uniformity would increase variability, allow larger batch sizes, and improve runtime flexibility.

6.3.3. Regularization Strategy

This work highlighted the potential of output regularization with linear decay for improving flux prediction. However, surface reconstructions remained limited, often collapsing to rigid, square-like structures. A promising next step is to combine this output regularization with more geometry-aware terms, such as curvature or edge constraints, and to evaluate these systematically. Doing so may reconcile the trade-off observed here between flux fidelity and surface accuracy.

6.3.4. Heliostat Position Strategy

Additionally, experiments with closer heliostat positions (5–15 m) demonstrated the benefit of using datasets that emphasize stronger flux detail. Future work should exploit this by introducing pretraining on closer, synthetic positions before fine-tuning on realistic configurations, thereby combining richer training signals with practical applicability.

6.3.5. Real-World Validation and Canting

Ultimately, the framework should be validated on real CSP heliostats to confirm its practical applicability. While simulated datasets have demonstrated promising performance—particularly in flux prediction—the step to real-world deployment requires addressing the simplified treatment of canting. Incorporating angular as well as translational components into canting would align simulated surfaces more closely with physical reality, ensuring that models trained on augmented data can transfer reliably to measured flux patterns.

With these recommendations in mind, this work concludes with a summary of its key findings, the limitations encountered, and the broader implications of the results for heliostat surface reconstruction using deep learning.

7

Conclusion

This work set out to explore the feasibility of reconstructing heliostat surfaces from flux density images using a deep learning model informed by the underlying physics of the system. The best-performing flux simulation model was able to reproduce flux density distributions with high accuracy, achieving an average flux image MAE of 1.55×10^{-2} on the test set—equivalent to a median accuracy of 84 % under the normalization used by Lewen et al. Although this is slightly below the 92 % median accuracy reported in their supervised approach, the results demonstrate that the method can generate flux predictions of competitive quality. At the same time, the corresponding surface reconstructions of this experiment were less reliable, highlighting the fundamentally ill-posed nature of the inverse problem.

To mitigate this ambiguity, several strategies were explored to reduce the solution space through regularization and dataset design. The strongest result was achieved when training on synthetic datasets with closer heliostat positions (5–15 m), yielding a median surface MAE of 2.4×10^{-4} on the test set. Nevertheless, when compared to the supervised approach by Lewen et al., which achieved a median surface MAE of 1.4×10^{-4}, the present method still falls short in terms of surface reconstruction accuracy.

Nonetheless, the surface predictions reveal that the network has internalized certain structural patterns, as the reconstructed surfaces partially resemble the mean ground truth surface of the training dataset. This suggests that the model is capable of capturing dataset-level biases, although it also does not reproduce the mean surface itself with full accuracy. More critically, its ability to generalize to individual heliostat geometries remains limited, as the reconstructions tend to collapse toward average-like shapes rather than faithfully representing unique surface variations.

Despite the limitations in surface reconstruction, the model retains significant functional value for CSP applications. By reproducing flux distributions with high fidelity, it can already serve as a surrogate tool for calibration or optimization tasks, where the goal is often to achieve a desired flux pattern rather than to recover the exact underlying geometry. In this sense, the network provides practically useful adjustments to heliostat surfaces without requiring direct access to ground truth measurements.

Beyond this functional utility, the main computational advantage of this framework lies in its ability to rapidly generate surface predictions via the integrated differentiable ray tracer. Once trained, the network can propose candidate surfaces much faster than conventional optimization- or raytracing-based reconstruction methods, making it suitable as a surrogate model in calibration or control scenarios. Importantly, the predicted surfaces are not unique: for most experiments, the model produces different geometries that reproduce the same flux distribution, thereby revealing more about the inherent ambiguity of the inverse problem. Only in the close-range (5–15 m) experiment did the predictions collapse toward an average-like surface, highlighting that this diversity depends on the input regime.

Looking ahead, a final advantage of this approach is that it does not inherently rely on surface measurements. Since the model can be supervised purely through flux images, future training could be carried

out directly on measured flux distributions, without the need for deflectometry data. This would significantly reduce the experimental effort required for dataset generation and make the method significantly easier to apply in operational CSP plants.

In summary, while precise surface reconstruction remains a challenge, this work demonstrates that physics-informed deep learning can deliver accurate flux predictions and capture some average structural surface behavior in the training data. It struggles however with the precise reconstruction of individual surfaces. With continued progress on regularizations and dataset design, this framework has potential to evolve into a practical tool for heliostat calibration and CSP field optimization.

References

[1] SolarPACES. *Shouhang Dunhuang Phase II 100 MW Tower*. https://solarpaces.nrel.gov/project/shouhang-dunhuang-phase-ii-100-mw-tower. Accessed: 2025-08-20. 2025.

[2] International Energy Agency. *Renewables 2024*. Licence: CC BY 4.0. Paris, 2024. URL: https://www.iea.org/reports/renewables-2024.

[3] Alfredo Peinado Gonzalo, Alberto Pliego Marugán, and Fausto Pedro García Márquez. "A review of the application performances of concentrated solar power systems". In: *Applied Energy* 255 (2019), p. 113893. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2019.113893. URL: https://www.sciencedirect.com/science/article/pii/S0306261919315806.

[4] Omar Behar, Abdallah Khellaf, and Kamal Mohammedi. "A review of studies on central receiver solar thermal power plants". In: *Renewable and Sustainable Energy Reviews* 23 (2013), pp. 12–39. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2013.02.017. URL: https://www.sciencedirect.com/science/article/pii/S1364032113001184.

[5] Naman Goyal, Akshansh Aggarwal, and Anil Kumar. "Concentrated solar power plants: A critical review of regional dynamics and operational parameters". In: *Energy Research Social Science* 83 (2022), p. 102331. ISSN: 2214-6296. DOI: https://doi.org/10.1016/j.erss.2021.102331. URL: https://www.sciencedirect.com/science/article/pii/S2214629621004230.

[6] International Energy Agency. *Concentrating Solar Power*. IEA Technology Roadmaps. Paris: OECD Publishing, 2010. DOI: 10.1787/9789264088139-en. URL: https://doi.org/10.1787/9789264088139-en.

[7] Lee A. Weinstein et al. "Concentrating Solar Power". In: *Chemical Reviews* 115.23 (2015), pp. 12797–12838. ISSN: 0009-2665. DOI: 10.1021/acs.chemrev.5b00397. URL: https://doi.org/10.1021/acs.chemrev.5b00397.

[8] Ugo Pelay et al. "Thermal energy storage systems for concentrated solar power plants". In: *Renewable and Sustainable Energy Reviews* 79 (2017), pp. 82–100. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2017.03.139. URL: https://www.sciencedirect.com/science/article/pii/S1364032117304021.

[9] Yabei Zhang et al. "Modeling the potential for thermal concentrating solar power technologies". In: *Energy Policy* 38.12 (2010). Special Section: Carbon Reduction at Community Scale, pp. 7884–7897. ISSN: 0301-4215. DOI: https://doi.org/10.1016/j.enpol.2010.09.008. URL: https://www.sciencedirect.com/science/article/pii/S0301421510006695.

[10] HeliosCSP. *Concentrated Solar Energy Plant*. Accessed: 2025-05-14. 2023. URL: https://helioscsp.com/integrating-concentrated-solar-power-and-photovoltaics-for-enhanced-energy-generation/.

[11] International Renewable Energy Agency (IRENA). *Renewable Power Generation Costs in 2023*. Tech. rep. Accessed: 2025-07-02. IRENA, Sept. 2024. URL: https://www.irena.org/-/media/Files/IRENA/Agency/Publication/2024/Sep/IRENA_Renewable_power_generation_costs_in_2023.pdf.

[12] Fraunhofer Institute for Solar Energy Systems ISE. *Levelized Cost of Electricity – Renewable Energy Technologies*. Tech. rep. Accessed: 2025-07-02. Fraunhofer ISE, Apr. 2024. URL: https://www.ise.fraunhofer.de/content/dam/ise/en/documents/publications/studies/EN2024_ISE_Study_Levelized_Cost_of_Electricity_Renewable_Energy_Technologies.pdf.

[13] Alberto Boretti and Stefania Castelletto. "Cost and performance of CSP and PV plants of capacity above 100 MW operating in the United States of America". In: *Renewable Energy Focus* 39 (2021), pp. 90–98. ISSN: 1755-0084. DOI: https://doi.org/10.1016/j.ref.2021.07.006. URL: https://www.sciencedirect.com/science/article/pii/S1755008421000417.

[14] David Feldman et al. *Winter 2025 Solar Industry Update*. Tech. rep. NREL/PR-7A40-93310. Accessed 2025-08-18. National Renewable Energy Laboratory (NREL), 2025. URL: https://www.nrel.gov/docs/fy25osti/93310.pdf.

[15] REN21. *Renewables 2024 Global Status Report collection, Renewables in Energy Supply*. https://www.ren21.net/gsr-2024/modules/energy_supply/02_market_and_industry_trends/02_csp/. Accessed: 2025-08-18. 2024.

[16] Solargis. *Solar resource map: Direct normal irradiation (DNI)*. https://solargis.com/resources/free-maps-and-gis-data. Prepared by Solargis for The World Bank Group and ESMAP. 2025.

[17] Sara Pascual, Pilar Lisbona, and Luis M. Romeo. "Thermal Energy Storage in Concentrating Solar Power Plants: A Review of European and North American RD Projects". In: *Energies* 15.22 (2022). ISSN: 1996-1073. DOI: 10.3390/en15228570. URL: https://www.mdpi.com/1996-1073/15/22/8570.

[18] O. Achkari and A. El Fadar. "Latest developments on TES and CSP technologies – Energy and environmental issues, applications and research trends". In: *Applied Thermal Engineering* 167 (2020), p. 114806. ISSN: 1359-4311. DOI: https://doi.org/10.1016/j.applthermaleng.2019.114806. URL: https://www.sciencedirect.com/science/article/pii/S1359431118363269.

[19] Alicia Crespo et al. "Latent thermal energy storage for solar process heat applications at medium-high temperatures – A review". In: *Solar Energy* 192 (2019). Thermal Energy Storage for Solar Applications, pp. 3–34. ISSN: 0038-092X. DOI: https://doi.org/10.1016/j.solener.2018.06.101. URL: https://www.sciencedirect.com/science/article/pii/S0038092X18306534.

[20] Alberto Boretti and Stefania Castelletto. "Trends in performance factors of large photovoltaic solar plants". In: *Journal of Energy Storage* 30 (2020), p. 101506. ISSN: 2352-152X. DOI: https://doi.org/10.1016/j.est.2020.101506. URL: https://www.sciencedirect.com/science/article/pii/S2352152X20300906.

[21] SolarPACES. *Jülich Solar Tower*. Accessed: 2025-05-14. 2021. URL: https://solarpaces.nrel.gov/project/julich-solar-tower.

[22] ARTIST Association. *PAINT Database*. Accessed: April 22, 2025. 2024. URL: https://paint-database.org.

[23] Deutsches Zentrum für Luft- und Raumfahrt (DLR). *Solartürme Jülich*. Accessed: 2025-05-15. 2024. URL: https://www.dlr.de/de/forschung-und-transfer/forschungsinfrastruktur/grossforschungsanlagen/solartuerme-juelich.

[24] J. Lewen et al. "Inverse Deep Learning Raytracing for heliostat surface prediction". In: *Solar Energy* 289 (2025), p. 113312. ISSN: 0038-092X. DOI: https://doi.org/10.1016/j.solener.2025.113312. URL: https://www.sciencedirect.com/science/article/pii/S0038092X25000751.

[25] Camilo A. Arancibia-Bulnes et al. "A survey of methods for the evaluation of reflective solar concentrator optics". In: *Renewable and Sustainable Energy Reviews* 69 (2017), pp. 673–684. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2016.11.048. URL: https://www.sciencedirect.com/science/article/pii/S136403211630747X.

[26] Sören Kammel and Fernando Puente Leon. "Deflectometric Measurement of Specular Surfaces". In: *IEEE Transactions on Instrumentation and Measurement* 57.4 (2008), pp. 763–769. DOI: 10.1109/TIM.2007.894185.

[27] Zhou Lu et al. "The Expressive Power of Neural Networks: A View from the Width". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/32cbf687880eb1674a07bf717761dd3a-Paper.pdf.

[28] Shuo Feng, Huiyu Zhou, and Hongbiao Dong. "Using deep neural network with small dataset to predict material defects". In: *Materials & Design* 162 (2019), pp. 300–310. ISSN: 0264-1275. DOI: https://doi.org/10.1016/j.matdes.2018.11.060. URL: https://www.sciencedirect.com/science/article/pii/S0264127518308682.

[29] Qizhe Xie et al. “Self-Training With Noisy Student Improves ImageNet Classification”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[30] Stanislav Pidhorskyi, Donald A. Adjeroh, and Gianfranco Doretto. “Adversarial Latent Autoencoders”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[31] Jostein Barry-Straume et al. “An Evaluation of Training Size Impact on Validation Accuracy for Optimized Convolutional Neural Networks”. In: *SMU Data Science Review* 1.4 (2018). URL: https://scholar.smu.edu/datasciencereview/vol1/iss4/12.

[32] Makoto M. Kelp, Christopher W. Tessum, and Julian D. Marshall. *Orders-of-magnitude speedup in atmospheric chemistry modeling through neural network-based emulation*. 2018. arXiv: 1808.03874 [physics.ao-ph]. URL: https://arxiv.org/abs/1808.03874.

[33] Steffen Ulmer et al. “Automated high resolution measurement of heliostat slope errors”. In: *Solar Energy* 85.4 (2011). SolarPACES 2009, pp. 681–687. ISSN: 0038-092X. DOI: https://doi.org/10.1016/j.solener.2010.01.010. URL: https://www.sciencedirect.com/science/article/pii/S0038092X1000023X.

[34] ARTIST consortium. *NURBS Tutorial — ARTIST Documentation*. https://artist.readthedocs.io/en/latest/nurbs_tutorial.html. Accessed: 2025-05-19. 2024.

[35] Anjana Deva Prasad et al. “NURBS-Diff: A Differentiable Programming Module for NURBS”. In: *Computer-Aided Design* 146 (2022), p. 103199. ISSN: 0010-4485. DOI: https://doi.org/10.1016/j.cad.2022.103199. URL: https://www.sciencedirect.com/science/article/pii/S0010448522000045.

[36] Tero Karras et al. *Analyzing and Improving the Image Quality of StyleGAN*. 2019. arXiv: 1912.04958 [cs.CV]. URL: http://arxiv.org/abs/1912.04958.

[37] ARTIST Association. *ARTIST: Open Source Project*. https://github.com/ARTIST-Association/ARTIST. Accessed: 2025-04-21. 2024.

[38] Jing Sun et al. *Physics-Driven Self-Supervised Deep Learning for Free-Surface Multiple Elimination*. 2025. arXiv: 2502.05189 [physics.geo-ph]. URL: https://arxiv.org/abs/2502.05189.

[39] J. Sun, L. Li, and L. Zhang. “Physics-Trained Neural Network as Inverse Problem Solver for Potential Fields: Downward Continuation between Arbitrary Surfaces”. In: 2025.1 (2025), pp. 1–5. ISSN: 2214-4609. DOI: https://doi.org/10.3997/2214-4609.2025101445. URL: https://www.earthdoc.org/content/papers/10.3997/2214-4609.2025101445.

[40] Xinmeng Luan et al. *Physics-Informed Neural Network-Driven Sparse Field Discretization Method for Near-Field Acoustic Holography*. 2025. arXiv: 2505.00897 [eess.AS]. URL: https://arxiv.org/abs/2505.00897.

[41] Z. Lu, M. Jin, S. Chen, et al. “Physics-driven self-supervised learning for fast high-resolution robust 3D reconstruction of light-field microscopy”. In: *Nature Methods* 22 (2025), pp. 1545–1555. DOI: 10.1038/s41592-025-02698-z. URL: https://doi.org/10.1038/s41592-025-02698-z.

[42] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. *Practical Bayesian Optimization of Machine Learning Algorithms*. 2012. arXiv: 1206.2944 [stat.ML]. URL: https://arxiv.org/abs/1206.2944.

[43] Saeed Rahmani et al. *A Systematic Review of Edge Case Detection in Automated Driving: Methods, Challenges and Future Directions*. 2024. arXiv: 2410.08491 [cs.RO]. URL: https://arxiv.org/abs/2410.08491.

[44] Jason Ansel 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. URL: https://docs.pytorch.org/assets/pytorch2-2.pdf.

[45] Anton Tenzler. *Heliostat Surface Prediction via Physics-Aware Deep Learning*. https://github.com/atenzler/Heliostat-Surface-Prediction-via-Physics-Aware-Deep-Learning-. Accessed: 2025-08-20. 2025.

A

Appendix

A.1. GitHub

All the main scripts and functions are uploaded to a GitHub Repository (https://github.com/atenzler/Heliostat_Surface_Prediction_via_Physics_Aware_Deep_Learning), with reduced example data to allow for inspection and reproducibility.

A.2. Sun Position Calculation

The following section provides a detailed description of the method used to compute the solar azimuth and elevation angles for a given time and observer location. The implementation is based on empirical astronomical models and includes corrections for atmospheric refraction and topocentric parallax [24].

Observer Inputs

The observer's geographic coordinates are given by:

- Latitude: ϕ in degrees
- Longitude: λ in degrees

These are converted to radians:

$$\phi_{\text{rad}} = \frac{\phi \cdot \pi}{180}, \quad \lambda_{\text{rad}} = \frac{\lambda \cdot \pi}{180}$$

Atmospheric parameters are assumed constant:

$$P = 1.01325\,\text{bar}, \quad T = 20^\circ\text{C}$$

1. Universal Time and Julian Day

Given time components $(\text{hour}, \text{minute}, \text{second})$, the universal time in hours is computed as:

$$UT = \text{hour} + \frac{\text{minute}}{60} + \frac{\text{second}}{3600}$$

The Julian Day offset JD_t from the reference epoch (J2000.0) is calculated as:

$$\text{If } m \leq 2: \quad y' = y - 1, \quad m' = m + 12$$

$$\text{Else:} \quad y' = y, \quad m' = m$$

$$JD_t = \lfloor 365.25(y' - 2000) \rfloor + \lfloor 30.6001(m' + 1) \rfloor + d + \frac{UT}{24} - 1158.5$$

where m is the month and y is the year integer.

Define:

$$t = JD_t + \frac{\Delta t}{86400} \quad \text{with} \quad \Delta t = 0$$

2. Heliocentric Longitude

The base angular term:

$$\theta = 0.0172019 \cdot t - 0.0563$$

The heliocentric longitude is computed using a harmonic expansion:

$$L = 1.740940 + 0.017202768683 \cdot t + 0.0334118 \cdot \sin(\theta) + 0.0003488 \cdot \sin(2\theta)$$

Add lunar perturbation:

$$L = L + 0.0000313 \cdot \sin(0.2127730 \cdot t - 0.585)$$

Add higher-order harmonic corrections:

$$\begin{aligned} L = L &+ 0.0000126 \cdot \sin(0.004243 \cdot t + 1.46) + 0.0000235 \cdot \sin(0.010727 \cdot t + 0.72) \\ &+ 0.0000276 \cdot \sin(0.015799 \cdot t + 2.35) + 0.0000275 \cdot \sin(0.021551 \cdot t - 1.98) \\ &+ 0.0000126 \cdot \sin(0.031490 \cdot t - 0.80) \end{aligned}$$

Apply correction for long-term orbital variation:

$$t_2 = \frac{t}{1000}$$

$$L = L + [((-0.000000230796 \cdot t_2 + 0.0000037976) \cdot t_2 - 0.000020458) \cdot t_2 + 0.00003976] \cdot t_2^2$$

3. Geocentric Coordinates

Apply a small nutation (small, periodic oscillation in the Earth's axial tilt and rotation axis) correction:

$$\Delta\psi = 0.0000833 \cdot \sin(0.0009252 \cdot t - 1.173)$$

Earth's obliquity:

$$\epsilon = -0.00000000621 \cdot t + 0.409086 + 0.0000446 \cdot \sin(0.0009252 \cdot t + 0.397)$$

Geocentric solar longitude:

$$\lambda_{\text{geo}} = L + \pi + \Delta\psi - 0.00009932$$

Right ascension α and declination δ:

$$\alpha = \arctan 2\left(\sin(\lambda_{\text{geo}}) \cdot \cos(\epsilon), \cos(\lambda_{\text{geo}})\right)$$

$$\delta = \arcsin\left(\sin(\epsilon) \cdot \sin(\lambda_{\text{geo}})\right)$$

4. Local Hour Angle

H represents the angular distance between the local meridian and the Sun's position in the sky, measured westward along the celestial equator. It is computed as:

$$H = 6.30038809903 \cdot JD_t + 4.8824623 + \Delta\psi \cdot 0.9174 + \lambda_{\text{rad}} - \alpha$$

5. Parallax Corrections

Apply parallax corrections to declination:

$$\Delta\delta = -0.0000426 \cdot (\sin(\phi_{\text{rad}}) - \delta \cdot \cos(\phi_{\text{rad}}))$$

$$\delta' = \delta + \Delta\delta$$

Adjusted components:

$$\begin{aligned} \cos\delta', \sin\delta' &= \cos(\delta'), \sin(\delta') \\ \cos H, \sin H &= \cos(H), \sin(H) \\ \Delta\alpha &= -0.0000426 \cdot \cos(\phi_{\text{rad}}) \cdot \sin(H) \\ \cos H', \sin H' &= \cos H + \Delta\alpha \cdot \sin H, \quad \sin H - \Delta\alpha \cdot \cos H \end{aligned}$$

6. Solar Elevation Angle

The solar elevation angle without refraction is:

$$\theta_{\text{elev, no refrac}} = \arcsin\left(\sin(\phi_{\text{rad}}) \cdot \sin\delta' + \cos(\phi_{\text{rad}}) \cdot \cos\delta' \cdot \cos H'\right)$$

Atmospheric refraction correction (if $\theta_{\text{elev, no refrac}} > -0.01$):

$$\Delta_{\text{refrac}} = \frac{0.084217 \cdot \frac{P}{1.01325}}{273 + T} \cdot \frac{1}{\tan\left(\theta_{\text{elev, no refrac}} + \frac{0.0031376}{\theta_{\text{elev, no refrac}} + 0.089186}\right)}$$

Final elevation angle:

$$\theta_{\text{elev}} = \theta_{\text{elev, no refrac}} + \Delta_{\text{refrac}}$$

Convert to degrees:

$$\theta^{\circ}_{\text{elev}} = \theta_{\text{elev}} \cdot \frac{180}{\pi}$$

7. Solar Azimuth Angle

The azimuth angle is computed as:

$$\theta_{\text{azim}} = -\arctan 2\left(\sin H', \cos H' \cdot \sin(\phi_{\text{rad}}) - \frac{\sin\delta'}{\cos\delta'} \cdot \cos(\phi_{\text{rad}})\right)$$

Convert to degrees:

$$\theta^{\circ}_{\text{azim}} = \theta_{\text{azim}} \cdot \frac{180}{\pi}$$

A.3. Number of Sun Positions

To verify the impact of sun position, experiments were conducted using 200 heliostats, comparing a setup with only one sun position per heliostat to one with eight sun positions. Using eight sun positions increases the input diversity, as the network receives eight flux images and corresponding sun positions for each heliostat instead of a single pair.

The results, shown in Table A.1, indicate that using multiple sun positions significantly reduces the MAE of the flux image predictions. This improvement is also visually evident in Figure A.1, where the predictions from the eight-sun setup more closely resemble the ground truth. Additionally, the surface MAE—used as an evaluation metric—also decreases when multiple sun positions are employed, confirming the advantage of using eight sun positions.

Table A.1: Comparison of flux image MAE and surface MAE when using one versus eight sun positions per heliostat. Incorporating multiple sun positions reduces both error metrics, highlighting the benefit of increased input diversity.

Sun Position	Flux image MAE	Surface MAE
One Sun Position	8.10×10^{-3}	3.78×10^{-4}
Eight Sun Positions	6.54×10^{-3}	3.42×10^{-4}

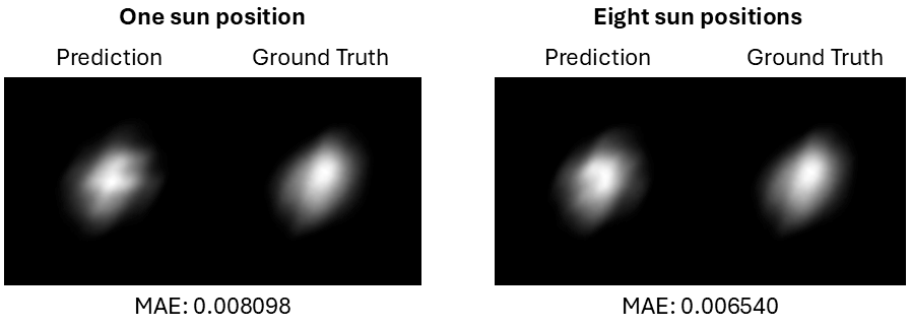


Figure A.1: Example predictions for one and eight sun positions per heliostat. Predictions from the eight-sun setup more closely resemble the ground truth flux distributions and achieve lower MAE values than the single-sun setup.

A.4. Raytracing Procedure

This section gives a detailed description of the raytracing procedure. The ray tracer begins by aligning each heliostat. To achieve this, it computes an orientation matrix that ensures the incoming solar rays are reflected toward the specified aim point. This orientation matrix is subsequently applied to all surface points and surface normals of the heliostat. The alignment follows the law of reflection:

$$\vec{r} = \vec{i} - 2(\vec{i} \cdot \vec{n})\vec{n} \tag{A.1}$$

where $\vec{r}$ is the reflected ray, $\vec{i}$ is the incident ray, and $\vec{n}$ is the surface normal. Here, $\cdot$ denotes the dot product. For the alignment procedure, $\vec{n}$ must be reoriented to achieve the correct reflection direction. The desired surface normal, $\vec{n}_{des}$, can be derived by inverting the law of reflection:

$$\vec{n}_{des} = \frac{-\vec{i} + \vec{r}_{des}}{\|-\vec{i} + \vec{r}_{des}\|} \tag{A.2}$$

where $\vec{r}_{des}$ is the unit vector pointing from the heliostat to the aim point, defined as:

$$\vec{r}_{des} = \frac{\vec{a} - \vec{p}}{\|\vec{a} - \vec{p}\|} \tag{A.3}$$

with $\vec{a}$ representing the aim point and $\vec{p}$ the position of the heliostat surface origin.

After determining $\vec{r}_{des}$, the desired surface normal $\vec{n}_{des}$ is calculated. The alignment process then iteratively solves for the two joint angles of the heliostat: one angle rotates about a primary axis, while the other rotates about an axis perpendicular to it. The joint angles are adjusted until the surface orientation matches $\vec{n}_{des}$. Once the correct angles are found, they are applied, and the heliostat is aligned with the chosen aim point.

The ray tracer then utilizes all the provided information to perform the ray tracing process. It first applies the law of reflection (Equation A.4) to compute the ideal reflection direction of rays for each surface point. In the next step, small angular deviations are introduced around this ideal direction to simulate light scattering. This results in a cloud of rays distributed around the preferred reflection direction.

Finally, the raytracer uses all the given information to trace the rays. It does this by applying the law of reflection in Equation A.4 again to compute a preferred reflection direction of rays for each surface point. In the next step, small angular deviations are applied to this reflection, to simulate the scattering of light. This gives a cloud of rays around the ideal reflection direction, to imitate non-ideal conditions.

The ray tracer then computes the line–plane intersection between the target area plane and each scattered ray to determine the exact impact points on the target area. These intersection points are projected onto a bitmap representation of the target area, which is divided into discrete pixels. Each ray impact increments the intensity value of the corresponding pixel, such that regions with multiple ray hits exhibit higher intensity values.

The overall output of the ray tracer is a bitmap with a predefined resolution and pixel size. In these images, brighter regions correspond to areas receiving higher ray intensity, while darker regions indicate areas receiving little to no rays. An example of this output was shown earlier in Figure 2.3 on the right side.

A.5. Hyperparameters

Table A.2: Search space for Bayesian hyperparameter optimization. Ranges are specified for learning parameters, weight decay, regularization strength, and scheduler settings, with conditional parameters applied depending on the selected scheduler.

Hyperparameter	**Search Space**
Optimizer	`["Adam"]`
Criterion	`["MAE", "MSE", "RMSE", "SSIM]`
Learning Rate	$[10^{-8}, 10^{-1}]$ (log-uniform)
Weight Decay	$[10^{-8}, 10^{-1}]$ (log-uniform)
Regularization on Outputs	$[10^{-6}, 10^{2}]$ (log-uniform)
Scheduler	`["None", "ReduceLROnPlateau, ExponentialLR, CyclicLR, CosineAnnealingLR]`
Scheduler Parameters (Conditional)	
ROP Factor	$[0.1, 0.9]$ (float)
ROP Patience	$[3, 10]$ (int)
ROP Threshold	$[10^{-6}, 10^{-3}]$ (log-uniform)
Exponential Gamma	$[0.85, 0.999]$ (float)
Cyclic Base LR	$[10^{-5}, 10^{-3}]$ (log-uniform)
Cyclic Max LR	$[10^{-3}, 10^{-1}]$ (log-uniform)
Cyclic Step Up	$[5, 50]$ (int)
Cosine T_max	$[10, 100]$ (int)
Cosine η_{min}	$[10^{-6}, 10^{-3}]$ (log-uniform)

Table A.3: Comparison of hyperparameter configurations obtained from the default grid search and Bayesian optimization. While both approaches use the Adam optimizer and 100 training epochs, the Bayesian search selects SSIM as the loss criterion and identifies a CyclicLR scheduler with tuned learning rate parameters and regularization values.

Hyperparameter	Grid Search Result	Bayesian Result
Optimizer	Adam	Adam
Criterion	MAE	SSIM
Batch Size	8	8
Max Epochs	100	100
Scheduler	None	CyclicLR
Learning Rate	0.001	1.2439×10^{-8}
Weight Decay	0	1.5708×10^{-5}
Output Weight Decay	-	60.993
Cyclic Base LR	-	5.0494×10^{-5}
Cyclic Max LR	-	1.0093×10^{-3}
Cyclic Step Up	-	11
Number of Inputs	variable	variable

A.6. Error Metric Evaluation

The choice of error metric in the training loop was evaluated. The error metric defines the magnitude of the loss between the predicted flux density image and the ground truth image, and therefore can significantly influence convergence and accuracy. Four models were trained with identical hyperparameters and 224 training samples, using MAE, MSE, RMSE, or SSIM as the loss function.

To enable a fair comparison, the predictions of all models were evaluated using a uniform “meta” metric—the MAE of the flux density image and the surface—even if another loss function (e.g., SSIM) was used during training. As shown in Table A.4, SSIM produced slightly better flux image predictions, while MAE and MSE achieved marginally better surface MAEs. RMSE performed the worst in both cases. Overall, to ensure comparability with previous work by Lewen et al., the MAE was chosen as the default error metric for all subsequent experiments.

Table A.4: Evaluation of different loss functions (MAE, MSE, RMSE, SSIM) in terms of flux density MAE and surface MAE. SSIM yielded the lowest flux density error, while MAE and MSE achieved the lowest surface errors. RMSE performed worst across both metrics.

Error Metric	Flux Density MAE	Surface MAE
MAE	1.48×10^{-2}	1.92×10^{-4}
MSE	1.42×10^{-2}	1.84×10^{-4}
RMSE	1.71×10^{-2}	2.13×10^{-4}
SSIM	1.26×10^{-2}	2.08×10^{-4}

A.7. Deflectometry Data Analysis

The 423 Deflectometry surfaces available in the PAINT database were analyzed in order to inform the regularizations. The detailed results of this analysis can be found here.

A.7.1. Order of Magnitude

Table A.5: Statistical summary of z-coordinates across 423 real heliostat surfaces from the PAINT database. Values characterize the order of magnitude and variability of surface deviations.

Statistic	Value [m]
Minimum z	-0.008906
Maximum z	0.007841
Mean μ_z	-0.000023
Mean $\lvert\mu_z\rvert$	0.000370
Standard Deviation σ_z	0.000518
Median	0.000009
Interquartile Range (IQR)	0.000560
Range $(z_{\max} - z_{\min})$	0.016747

A.7.2. Curvature

Table A.6: Statistical summary of Laplacian curvature values across all facets and surfaces in the PAINT database. The results capture the distribution and range of curvature magnitudes used to inform regularization design.

Statistic	Value [1/m]
Minimum C	-0.005466
Maximum C	0.100079
Mean μ_C	0.020191
Mean $\lvert\mu_C\rvert$	0.020438
Standard Deviation σ_C	0.024946
Median	0.000183
Interquartile Range (IQR)	0.040686
Range ($C_{\max} - C_{\min}$)	0.105544

Table A.7: Statistical comparison of Laplacian curvature between edge and center control points across all facets. Edge regions show substantially higher curvature magnitudes than central regions.

Statistic	Edge C [1/m]	Center C [1/m]
Minimum C	0.028818	0.000000
Maximum C	0.100079	0.005466
Mean μ_C	0.046355	0.000281
Mean $\lvert\mu_C\rvert$	0.046355	0.000281
Standard Deviation σ_C	0.014326	0.000307
Median	0.040710	0.000187
Interquartile Range (IQR)	0.001350	0.000285
Range ($C_{\text{max}} - C_{\text{min}}$)	0.071261	0.005466

A.7.3. Tilt

Table A.8: Distribution of facet tilt angles relative to the vertical axis. Tilt angles are generally small, with most facets exhibiting only slight deviations.

Statistic	Value [°]
Mean tilt angle	0.0428
Median tilt angle	0.0442
Standard Deviation	0.0343
Maximum tilt angle	0.4383
Minimum tilt angle	0.0000

A.7.4. Edge Behavior

Table A.9: Statistical summary of edge dip behavior across all surfaces, quantifying the tendency of facet edges to deviate relative to their interior regions.

Statistic	Value
Minimum	-0.003402
Maximum	0.002569
Mean	0.000220
Absolute Mean	0.000254
Standard Deviation σ	0.000246
Median	0.000202
Interquartile Range (Q3 - Q1)	0.000291
Range (max - min)	0.005971

A.8. Regularization Implementation

Table A.10: Search space for Bayesian optimization of regularization hyperparameters. Each regularization term was explored across several orders of magnitude using log-uniform sampling.

Hyperparameter	Search Space
Limit z-values, weight	$[10^2, 10^6]$ (log-uniform)
Curvature Penalty, weight	$[10^3, 10^7]$ (log-uniform)
Curvature Penalty, ref. curvature $[1/m]$	$[10^{-5}, 10^{-3}]$ (log-uniform)
Edge Dip, weight	$[10^2, 10^6]$ (log-uniform)
Edge Dip, dip margin [m]	$[10^{-5}, 10^{-3}]$ (log-uniform)
Tilt Penalty, weight	$[10^2, 10^5]$ (log-uniform)
Regularization on outputs, weight	$[10^{-2}, 10^2]$ (log-uniform)

Table A.11: Comparison of regularization parameter values obtained from surface analysis (V1) and Bayesian optimization (V2). The Bayesian search produced different weighting scales and margins across terms, reflecting alternative trade-offs between geometric constraints.

Regularization Term	V1 (Surface Analysis)	V2 (Bayesian Optimization)
Limit z-values, weight	1×10^5	2.73×10^5
Limit z-values, limit [m]	5.0×10^{-4}	5.0×10^{-4}
Curvature Penalty, weight	1×10^6	1.12×10^3
Curvature Penalty, ref. curvature $[1/m]$	5.0×10^{-5}	9.52×10^{-4}
Edge Dip, weight	1×10^5	8.20×10^2
Edge Dip, dip margin [m]	2.2×10^{-4}	1.30×10^{-5}
Tilt Penalty, weight	1×10^4	1.67×10^3
Regularization on outputs, weight	–	8.47×10^1

Table A.12: Additional regularization terms explored individually. Parameters specify the decay strategy for output regularization and the thresholds applied in repulsion regularization.

Regularization Term	Weight	Parameters
Regularization on Outputs, half	100	`linear_decay = True` (decay to 0 by half epochs)
Regularization on Outputs, max	100	`linear_decay = True` (decay to 0 by max epochs)
Repulsion Regularization	1000	`repulsion_loss = 0.001` `repulsion_threshold = 0.004`

A.9. Additional Results

This section is structured to show all results form Table 5.1, and is ordered in the same way as the table.

A.9.1. Baseline Results

Standard

Figure A.2 and Figure A.3 show the surface predictions and the flux density predictions of the standard experiment configuration respectively. These will be used as reference for all other experiments. All experiments below will follow this general result structure, and will not be captioned individually.

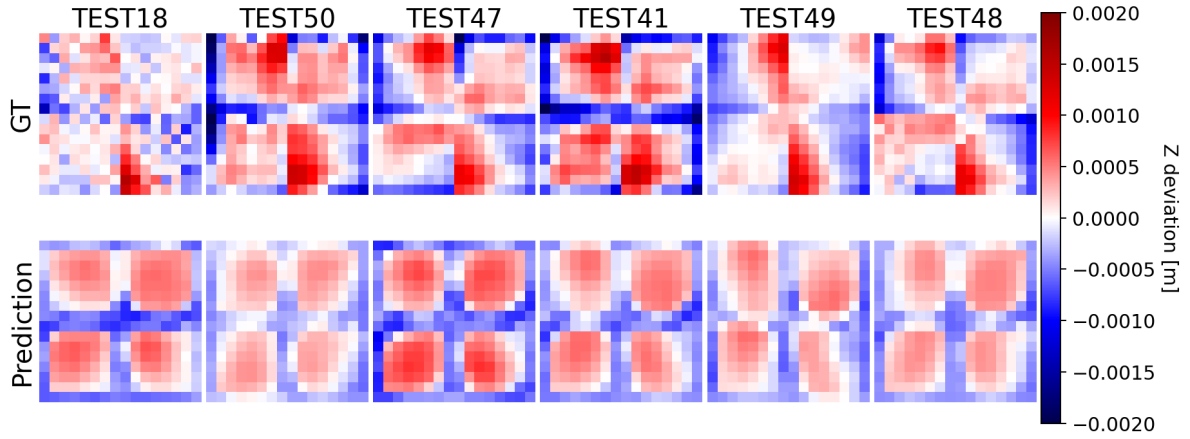
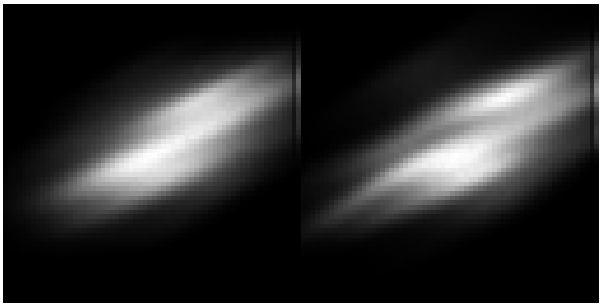
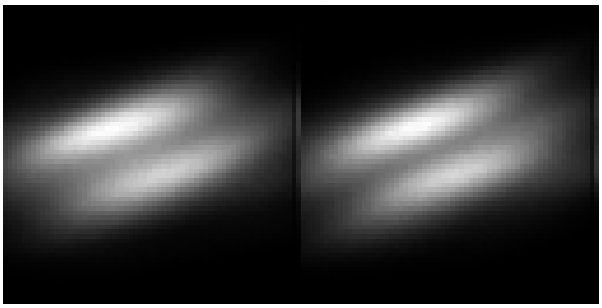


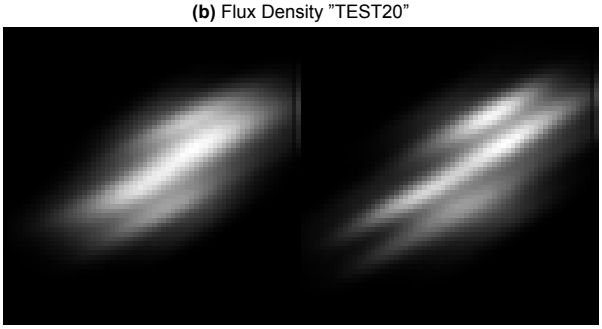
Figure A.2: Surface predictions of the standard model.



(a) Flux Density ”TEST2”



(b) Flux Density ”TEST20”



(c) Flux Density ”TEST52”

Figure A.3: Comparison of flux density predictions (left) and ground truth (right) of the standard model.

Surface model checkpoint (50k Training Samples)

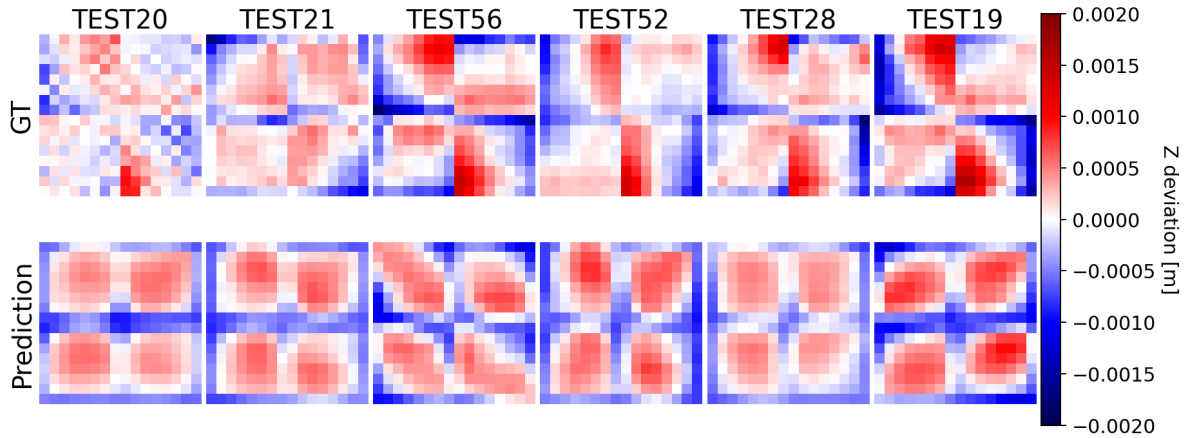
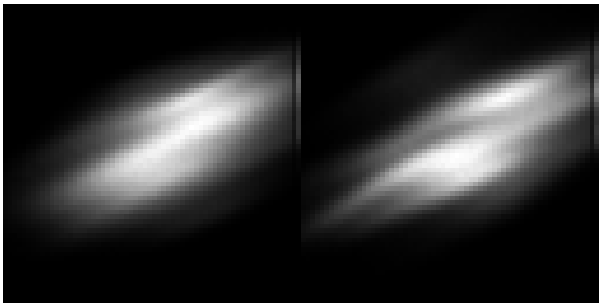
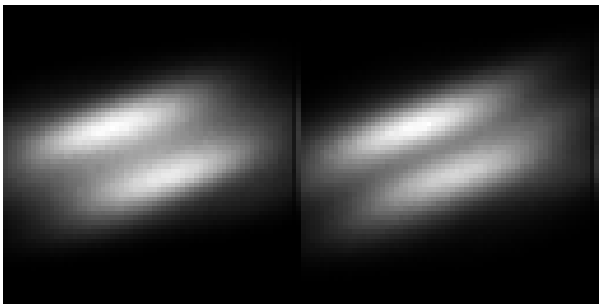


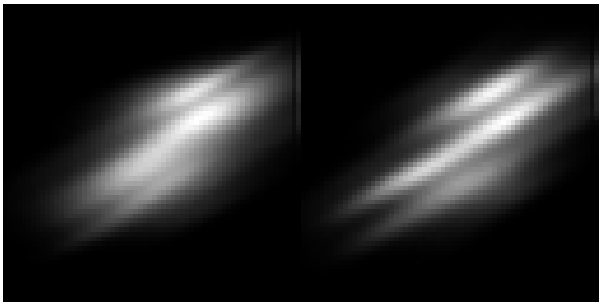
Figure A.4: Surface predictions



(a) Flux Density ”TEST2”



(b) Flux Density ”TEST20”



(c) Flux Density ”TEST52”

Figure A.5: Comparison of flux density predictions (left) and ground truth (right)

Figure A.6 shows the development of the surface MAE of predicitions during training of the model.

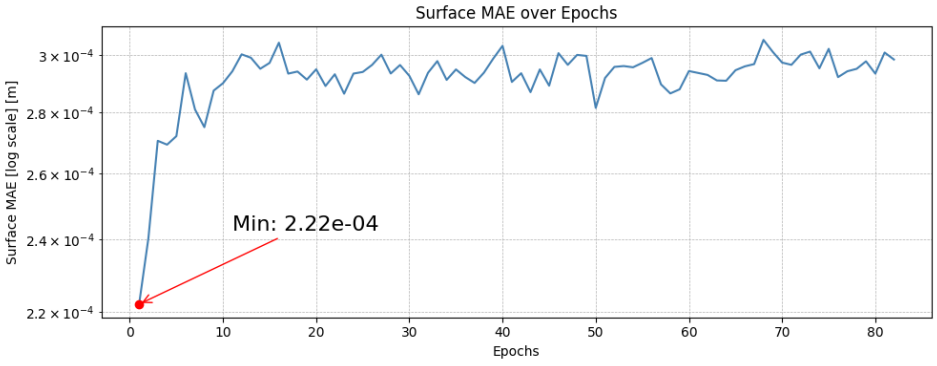


Figure A.6: Surface MAE curve during training of the baseline 50k inputs experiments

Train model checkpoint (50k Training Samples)

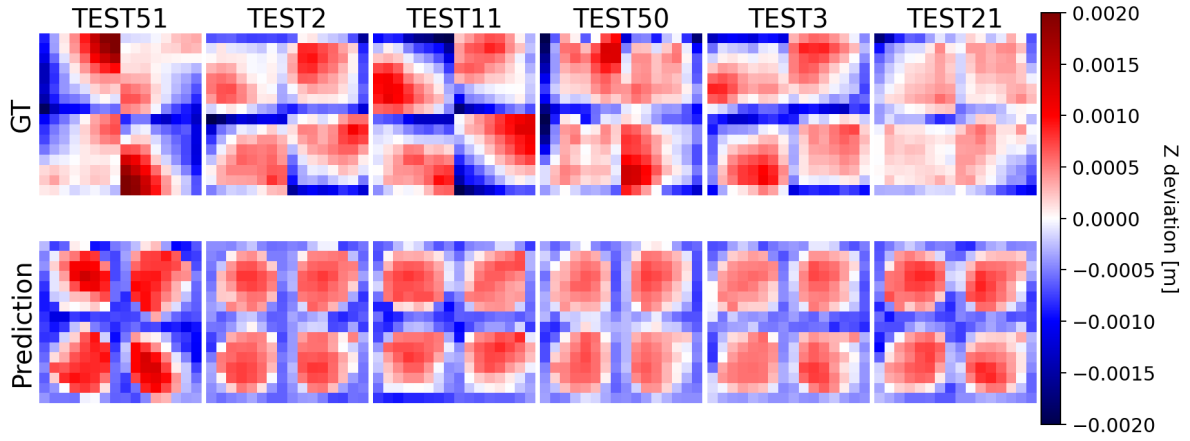
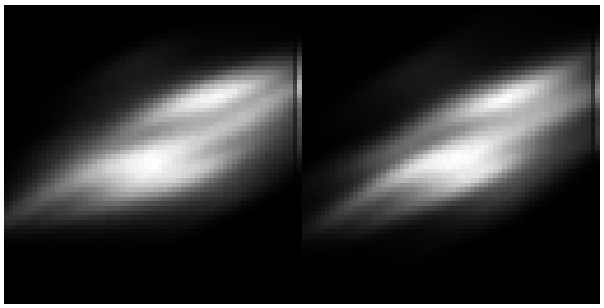
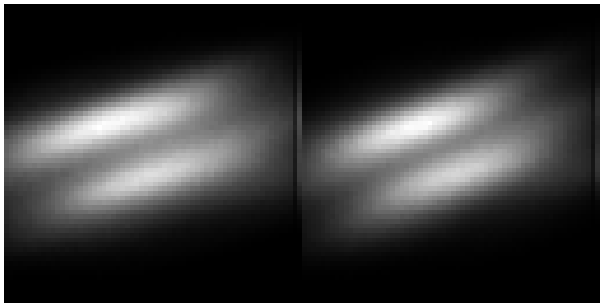


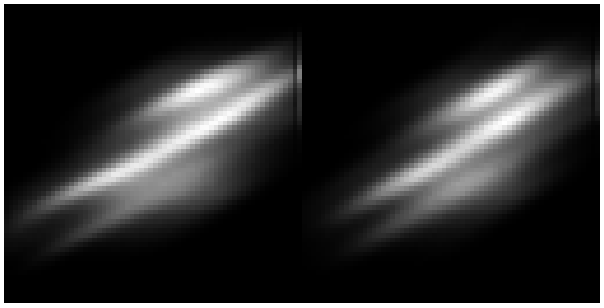
Figure A.7: Surface predictions



(a) Flux Density "TEST2"



(b) Flux Density "TEST20"



(c) Flux Density "TEST52"

Figure A.8: Comparison of flux density predictions (left) and ground truth (right)

A.9.2. Loss Function

SSIM

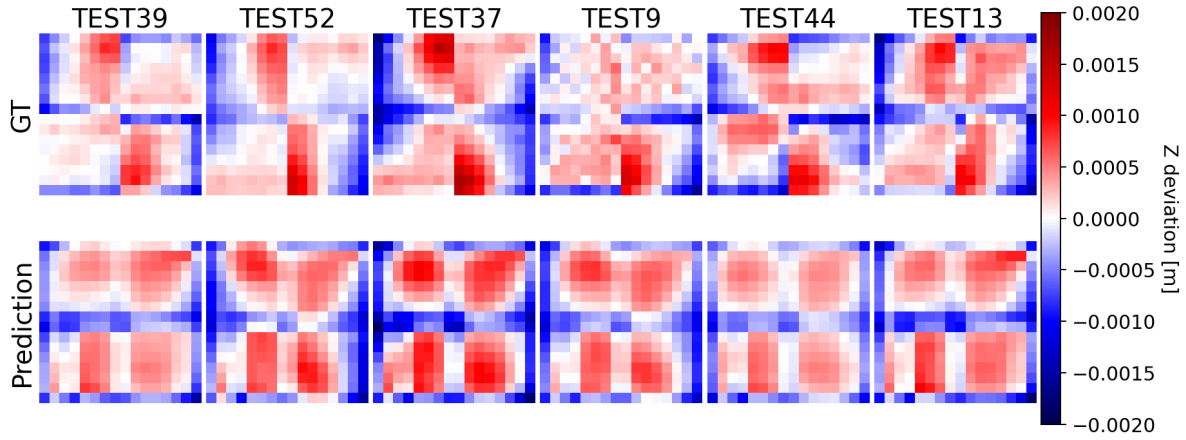
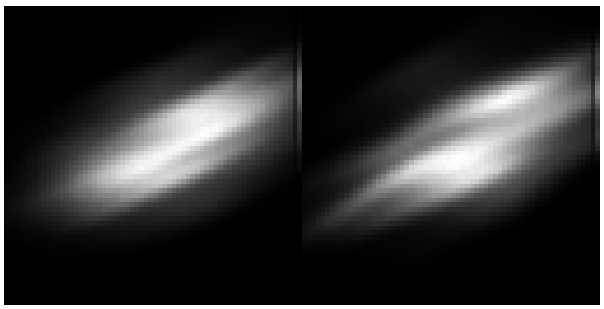
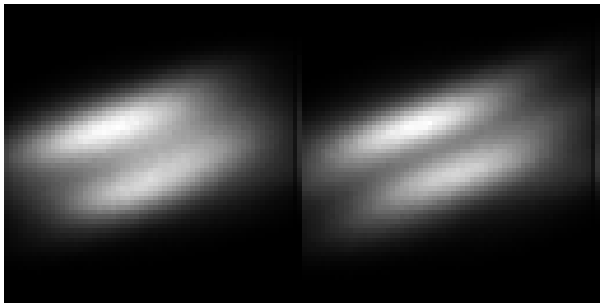


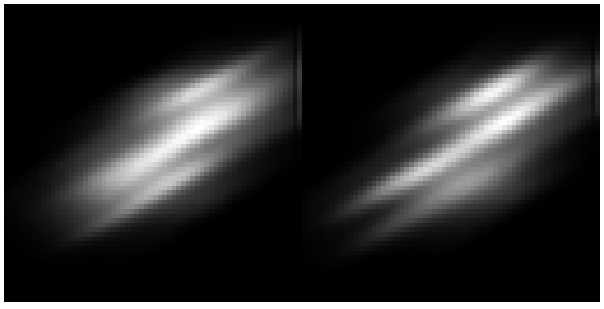
Figure A.9: Surface predictions



(a) Flux Density "TEST2"



(b) Flux Density "TEST20"



(c) Flux Density "TEST52"

Figure A.10: Comparison of flux density predictions (left) and ground truth (right)

A.9.3. Input Scaling

10k inputs

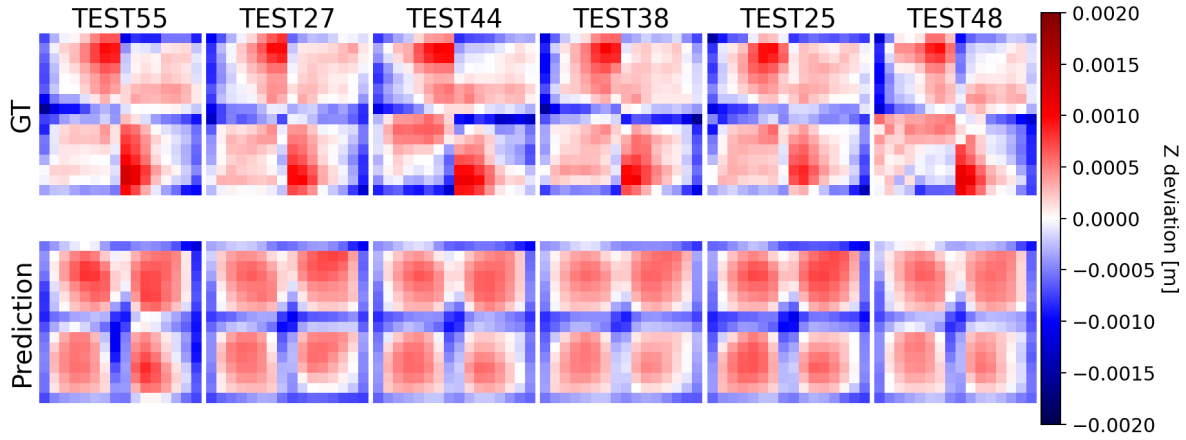
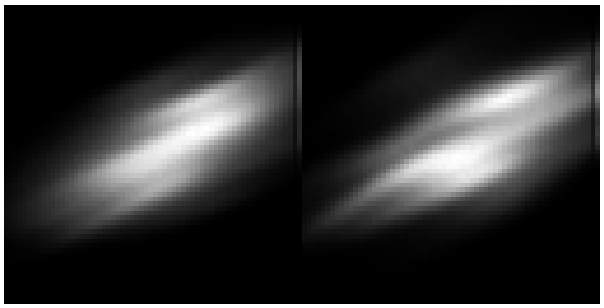
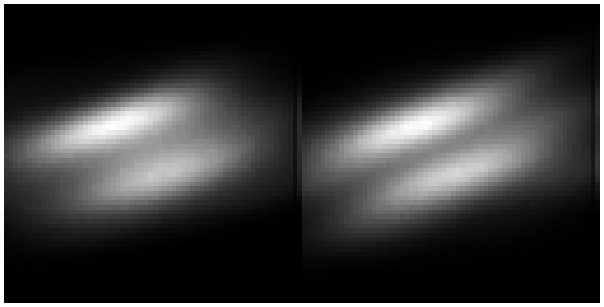


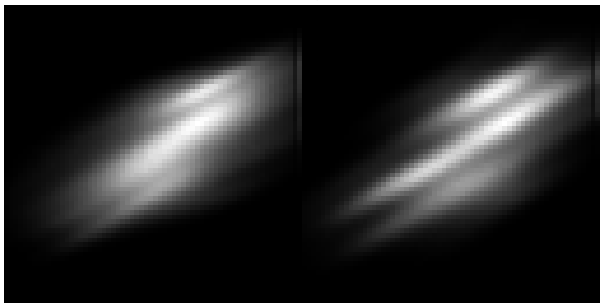
Figure A.11: Surface predictions



(a) Flux Density "TEST2"



(b) Flux Density "TEST20"



(c) Flux Density "TEST52"

Figure A.12: Comparison of flux density predictions (left) and ground truth (right)

A.9.4. Hyperparameters

Bayesian opt.

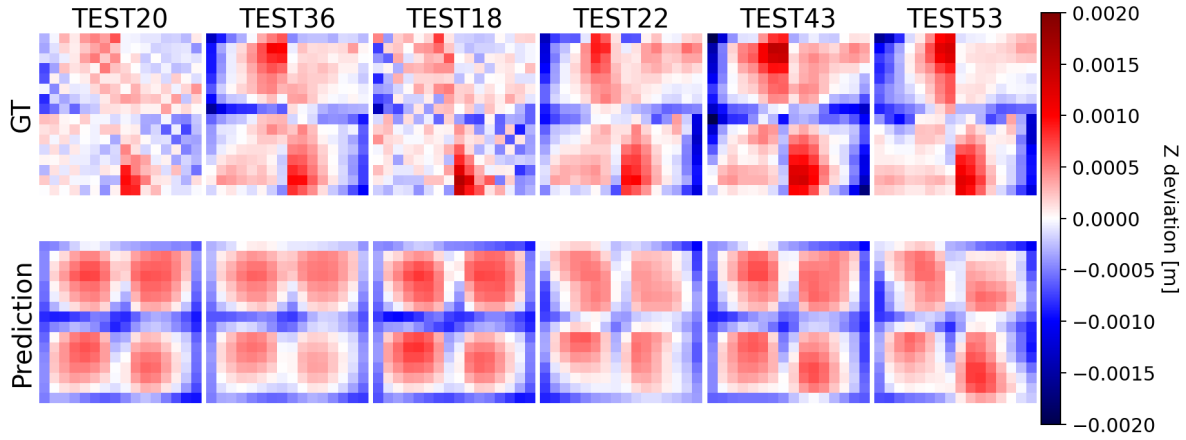
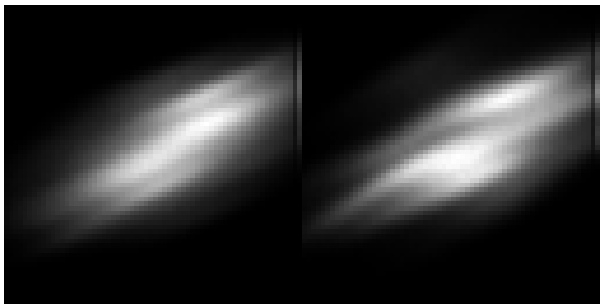
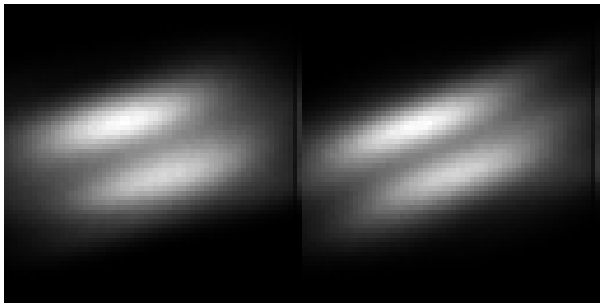


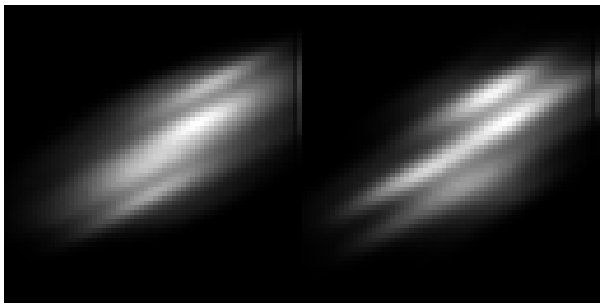
Figure A.13: Surface predictions



(a) Flux Density ”TEST2”



(b) Flux Density ”TEST20”



(c) Flux Density ”TEST52”

Figure A.14: Comparison of flux density predictions (left) and ground truth (right)

A.9.5. Regularization Results

All regs V1 5k

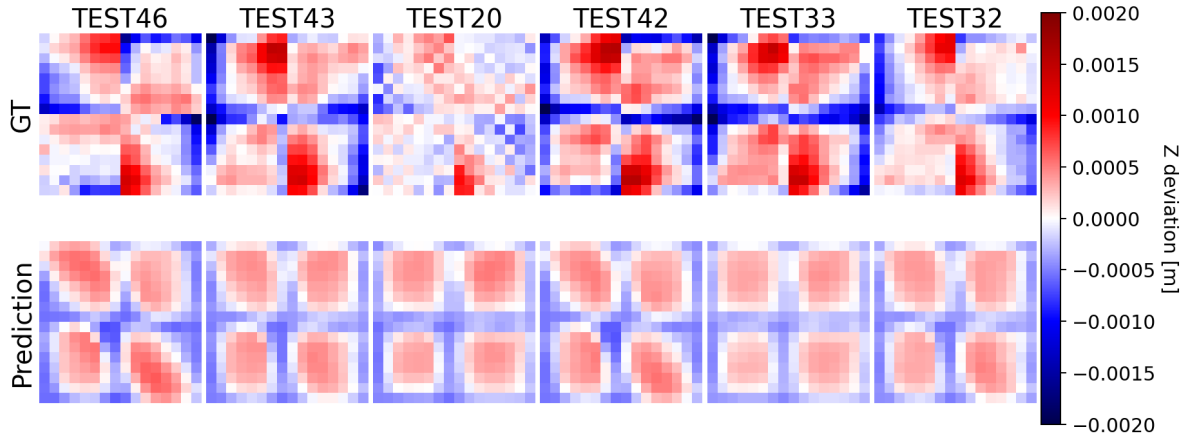
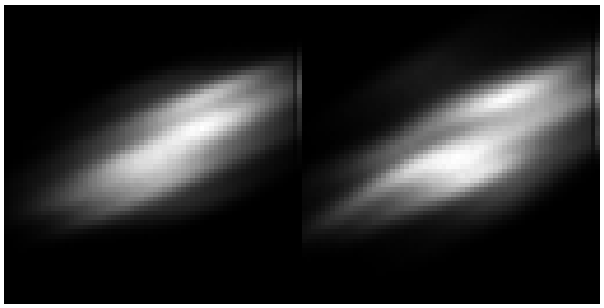
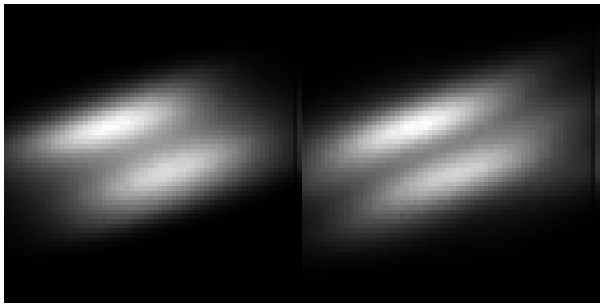


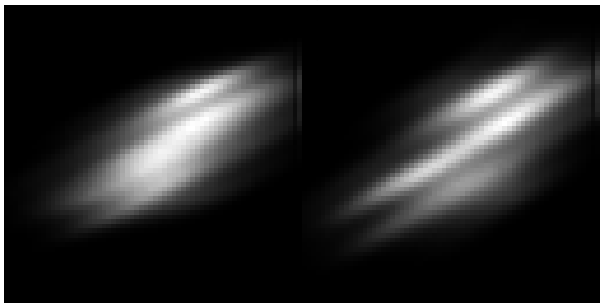
Figure A.15: Surface predictions



(a) Flux Density "TEST2"



(b) Flux Density "TEST20"



(c) Flux Density "TEST52"

Figure A.16: Comparison of flux density predictions (left) and ground truth (right)

All regs V1 50k

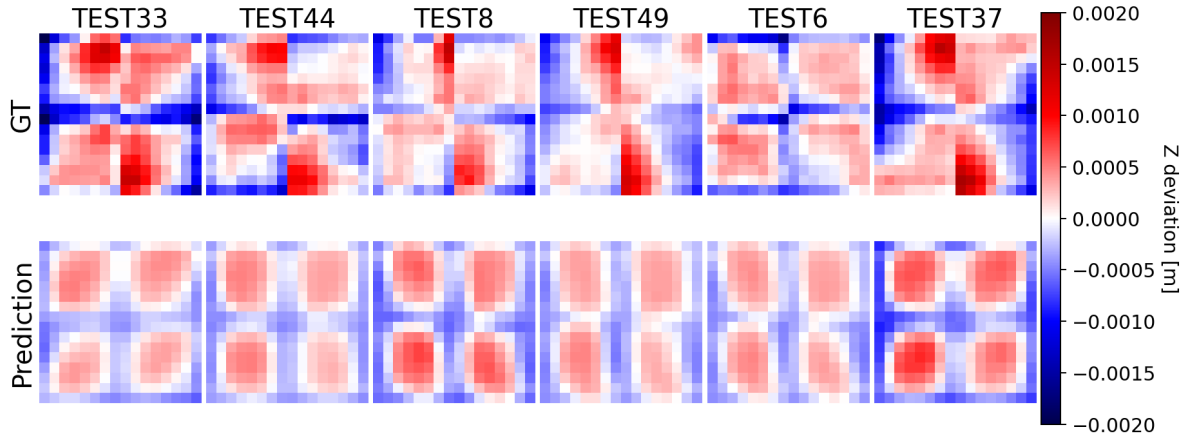
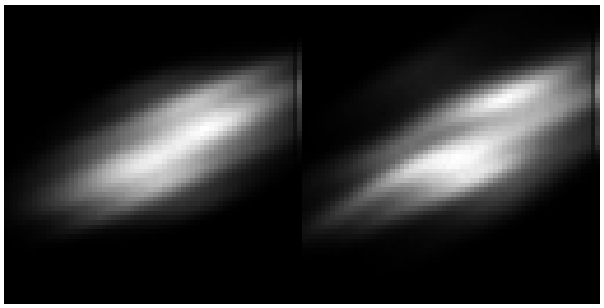
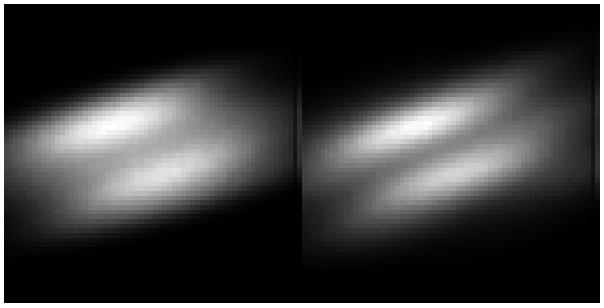


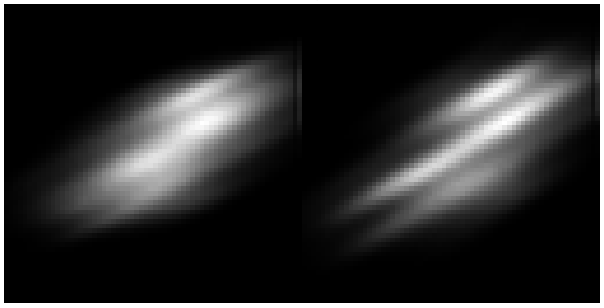
Figure A.17: Surface predictions



(a) Flux Density "TEST2"



(b) Flux Density "TEST20"



(c) Flux Density "TEST52"

Figure A.18: Comparison of flux density predictions (left) and ground truth (right)

All regs V2 (Bayes)

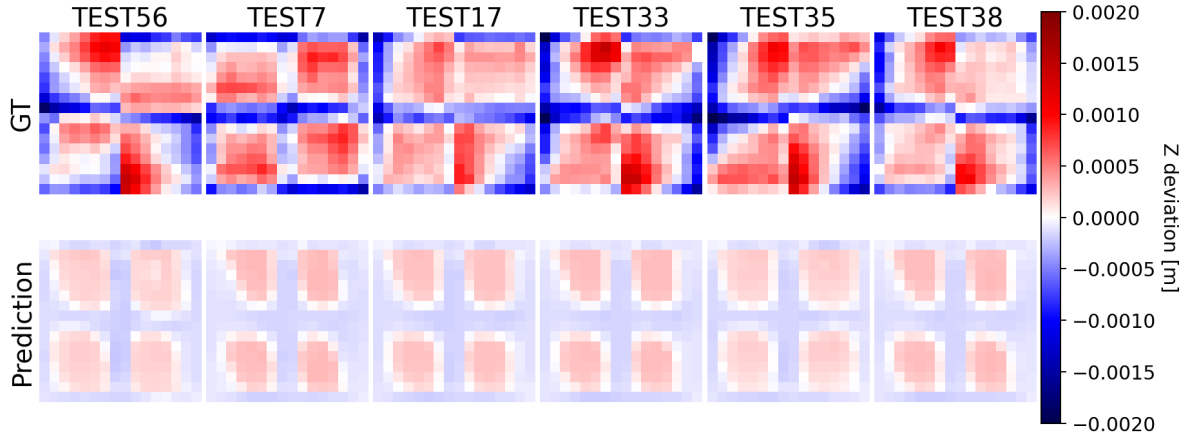
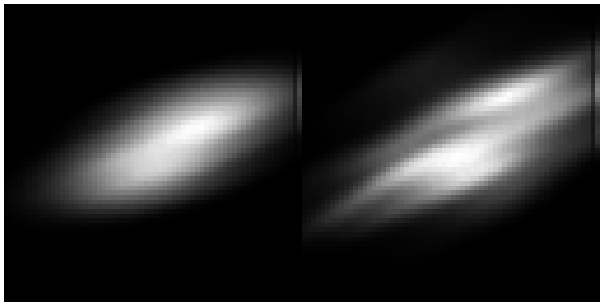
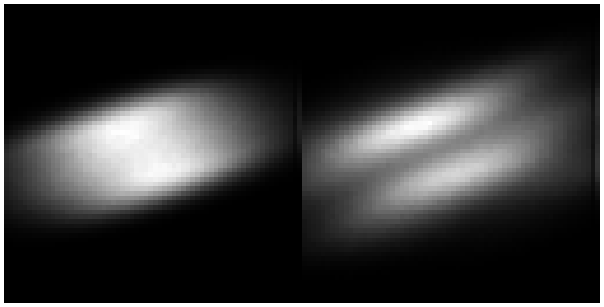


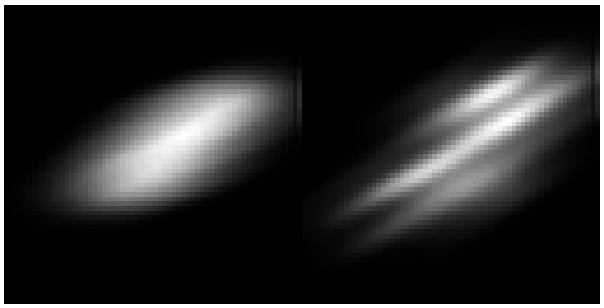
Figure A.19: Surface predictions



(a) Flux Density "TEST2"



(b) Flux Density "TEST20"



(c) Flux Density "TEST52"

Figure A.20: Comparison of flux density predictions (left) and ground truth (right)

Figure A.21 illustrates the surface MAE during the progression of training the V2 regularization configuration.

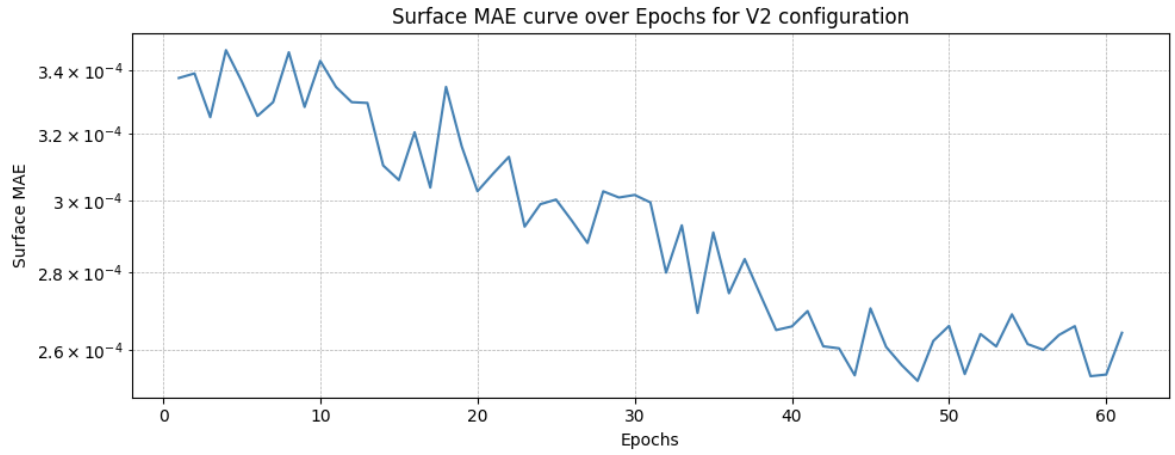


Figure A.21: Surface MAE during training of the V2 configuration.

Repulsion 200

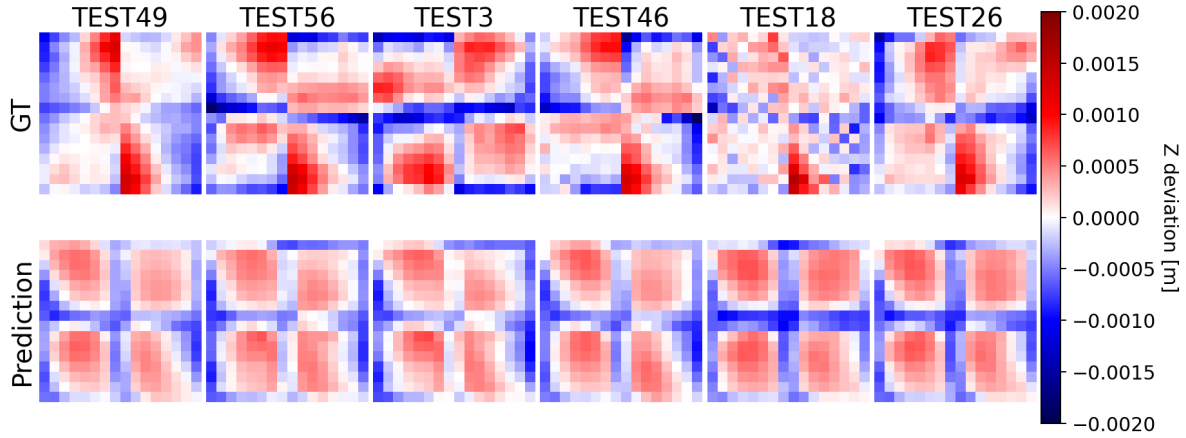
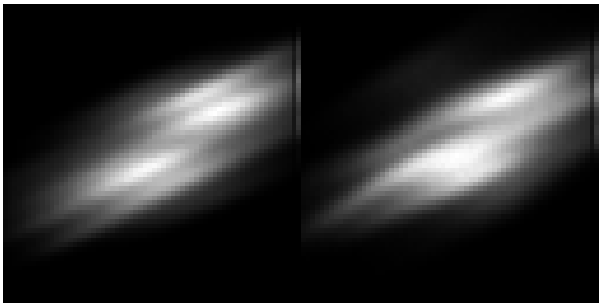
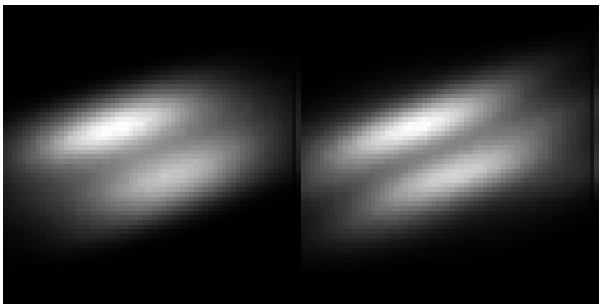


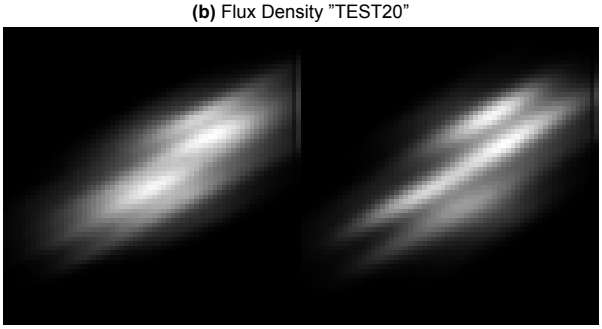
Figure A.22: Surface predictions



(a) Flux Density ”TEST2”



(b) Flux Density ”TEST20”



(c) Flux Density ”TEST52”

Figure A.23: Comparison of flux density predictions (left) and ground truth (right)

Repulsion 5k

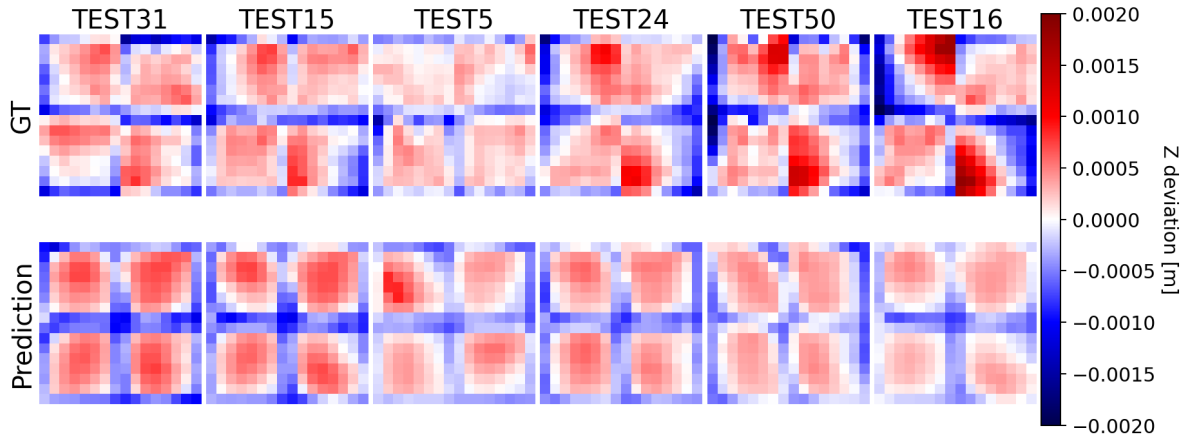
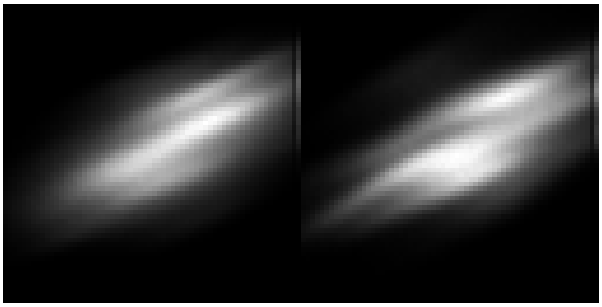
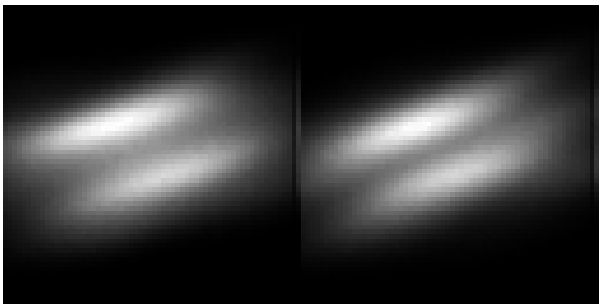


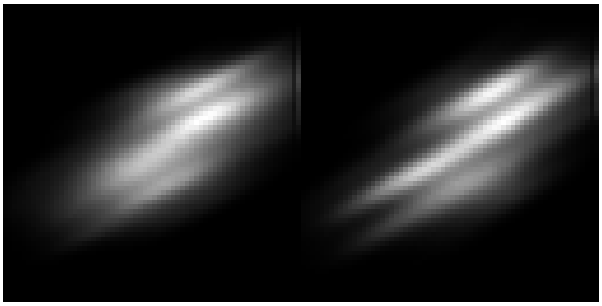
Figure A.24: Surface predictions



(a) Flux Density ”TEST2”



(b) Flux Density ”TEST20”



(c) Flux Density ”TEST52”

Figure A.25: Comparison of flux density predictions (left) and ground truth (right)

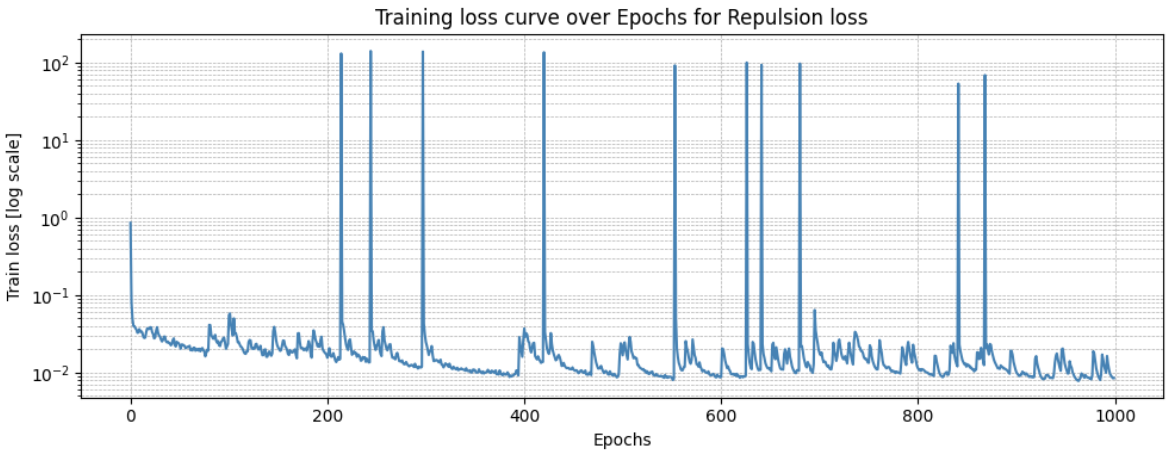


Figure A.26: Repulsion regularization training loss curve over 1000 epochs.

Output reg. lin decay, max epochs

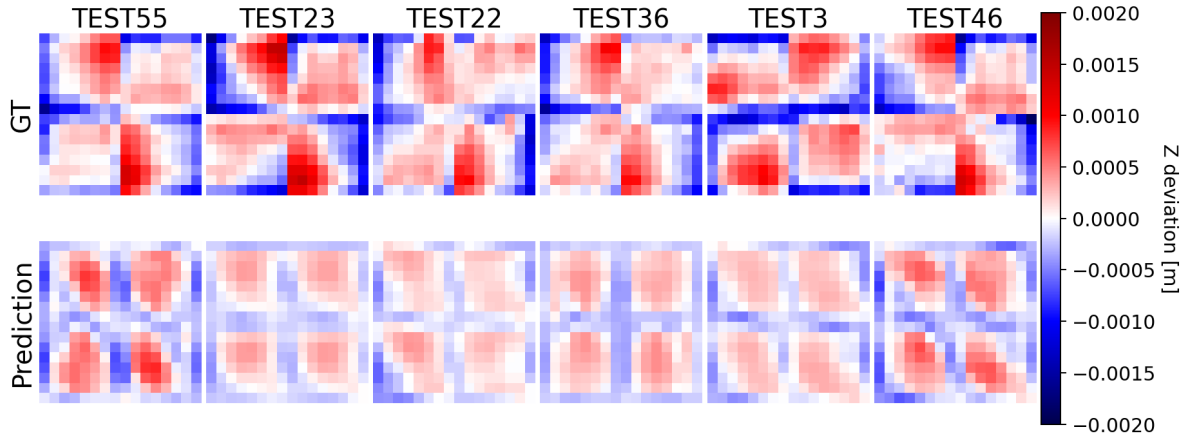
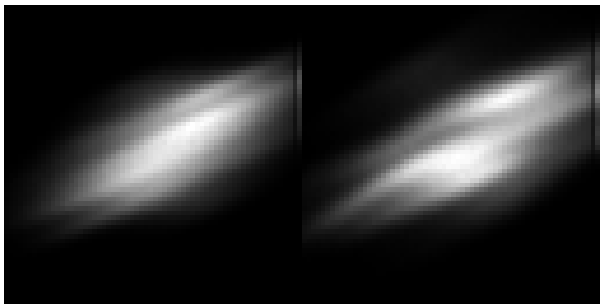
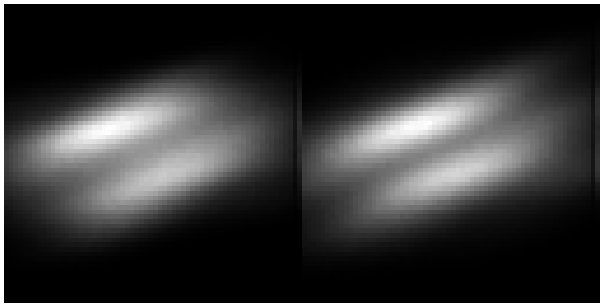


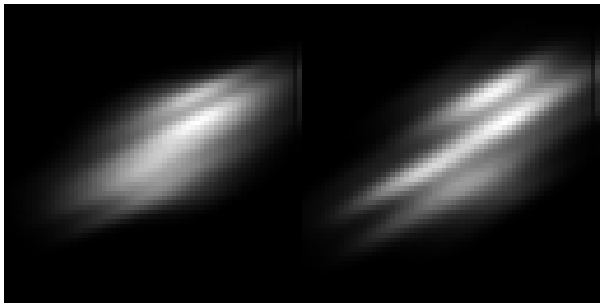
Figure A.27: Surface predictions



(a) Flux Density "TEST2"



(b) Flux Density "TEST20"



(c) Flux Density "TEST52"

Figure A.28: Comparison of flux density predictions (left) and ground truth (right)

Output reg. lin decay, half epochs, surface model

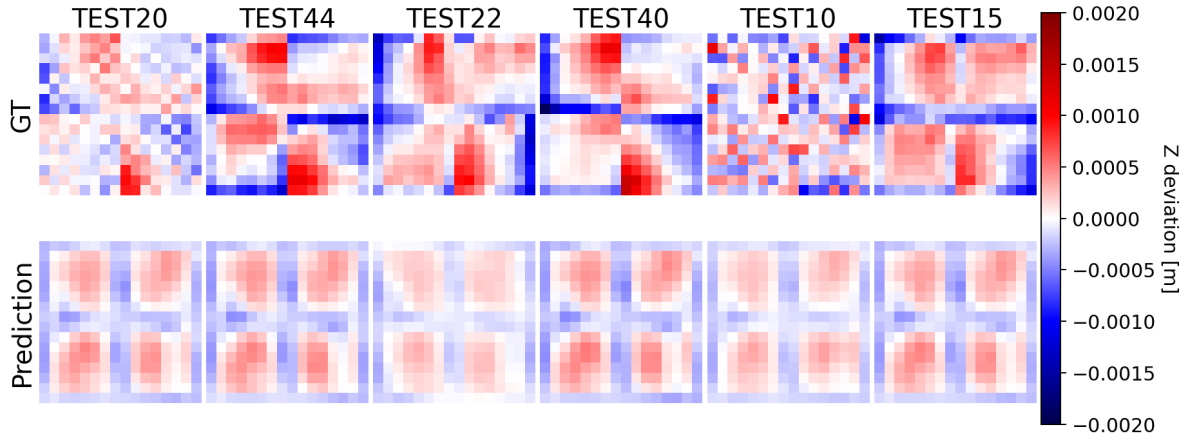
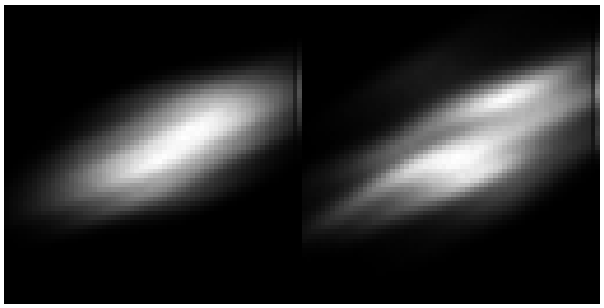
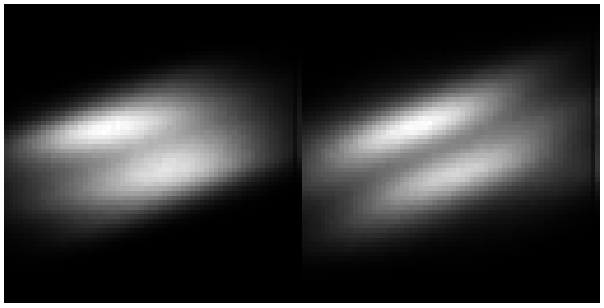


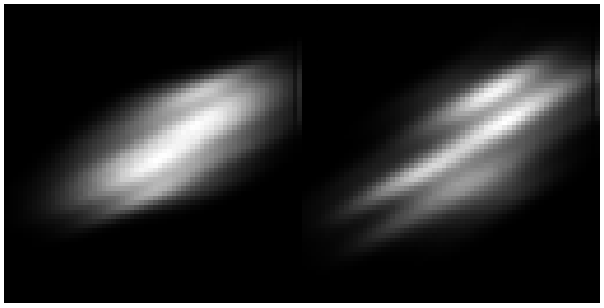
Figure A.29: Surface predictions



(a) Flux Density "TEST2"



(b) Flux Density "TEST20"



(c) Flux Density "TEST52"

Figure A.30: Comparison of flux density predictions (left) and ground truth (right)

Output reg. lin decay, half epochs, train model

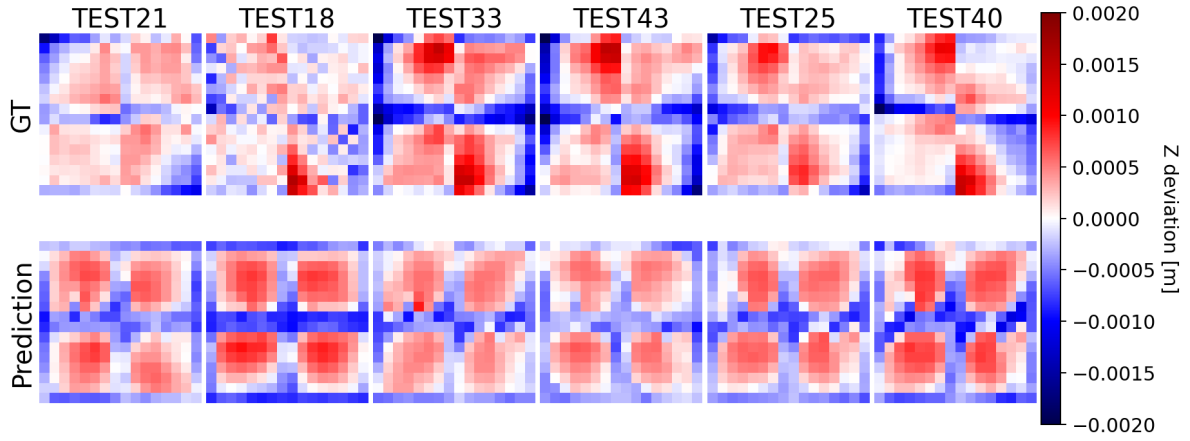
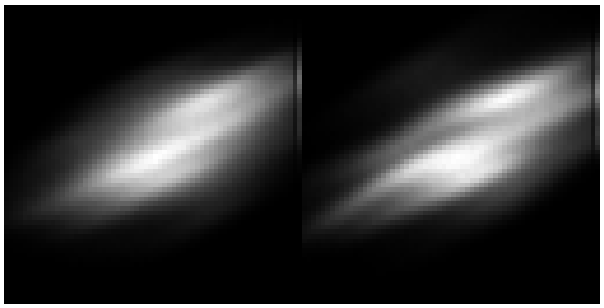
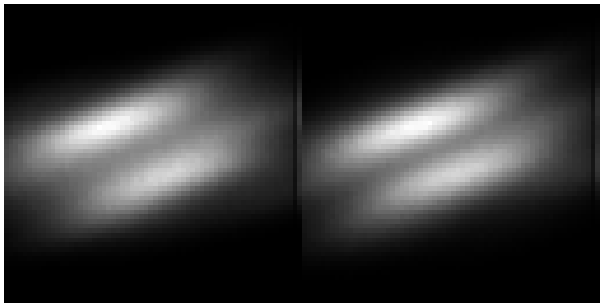


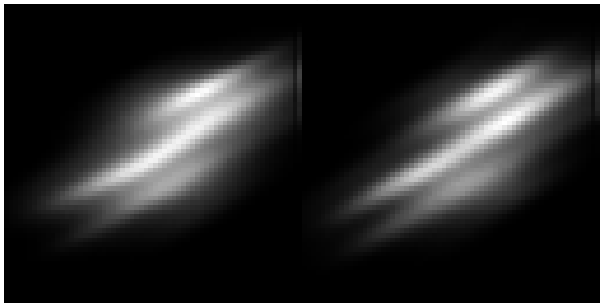
Figure A.31: Surface predictions



(a) Flux Density "TEST2"



(b) Flux Density "TEST20"



(c) Flux Density "TEST52"

Figure A.32: Comparison of flux density predictions (left) and ground truth (right)

A.9.6. Synthetic Dataset

Close, 30-70m

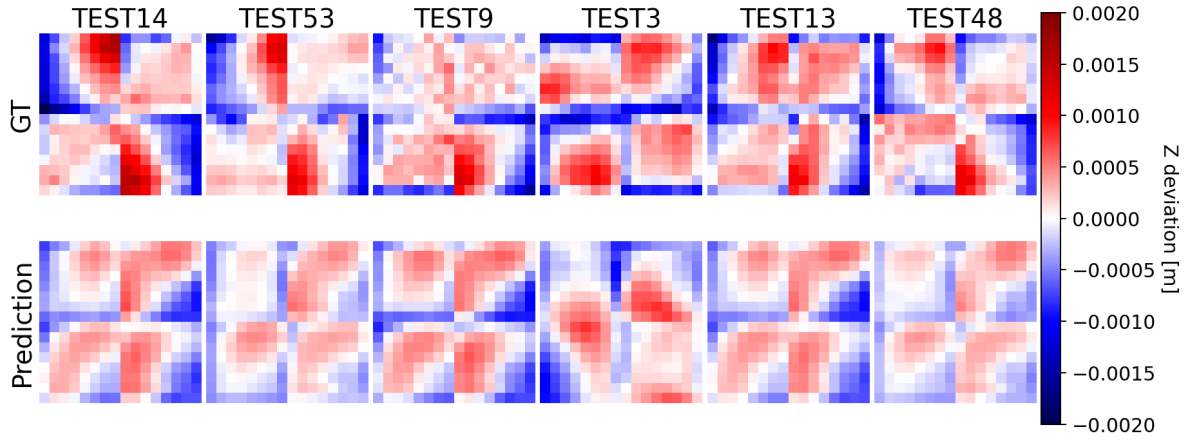
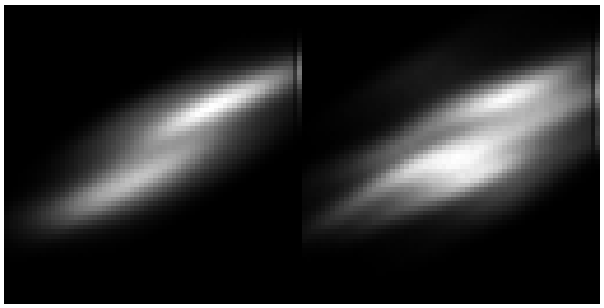
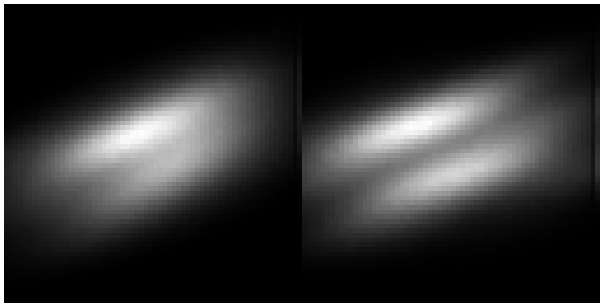


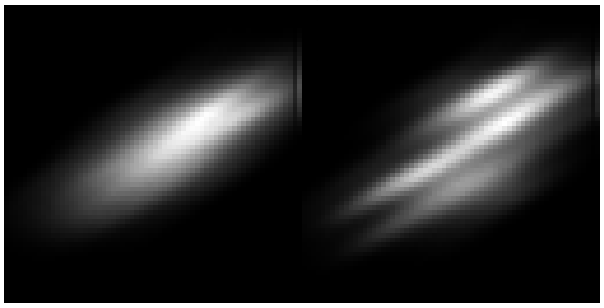
Figure A.33: Surface predictions



(a) Flux Density "TEST2"



(b) Flux Density "TEST20"



(c) Flux Density "TEST52"

Figure A.34: Comparison of flux density predictions (left) and ground truth (right)

Close, 30-70m, canting as input

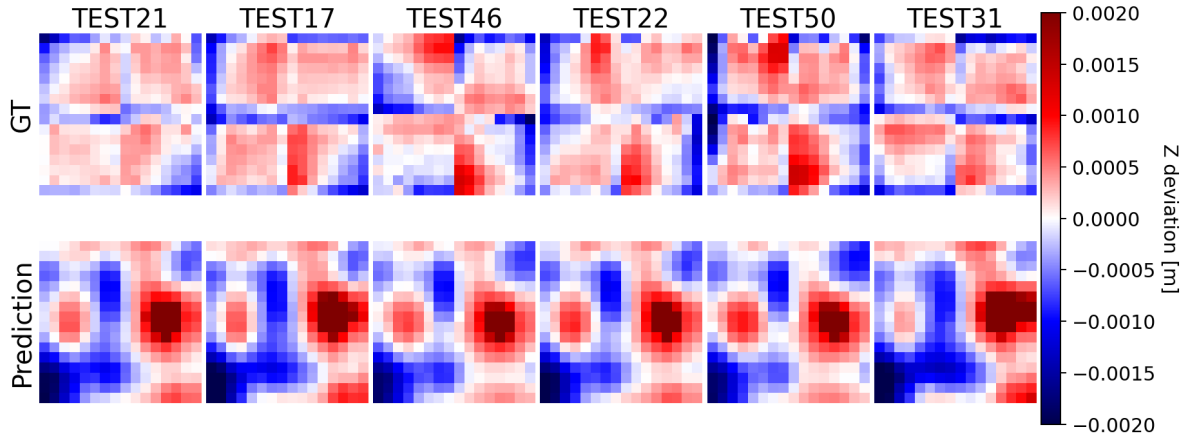
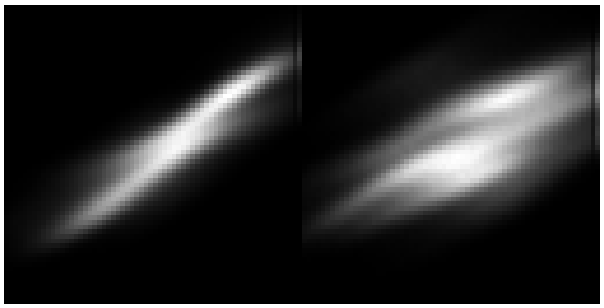
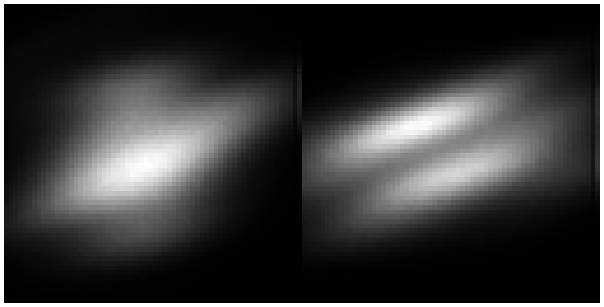


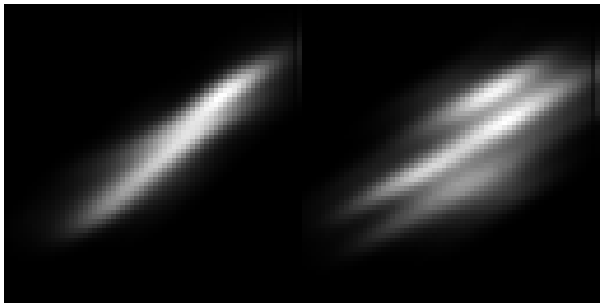
Figure A.35: Surface predictions



(a) Flux Density "TEST2"



(b) Flux Density "TEST20"



(c) Flux Density "TEST52"

Figure A.36: Comparison of flux density predictions (left) and ground truth (right)

Close, 5-15m

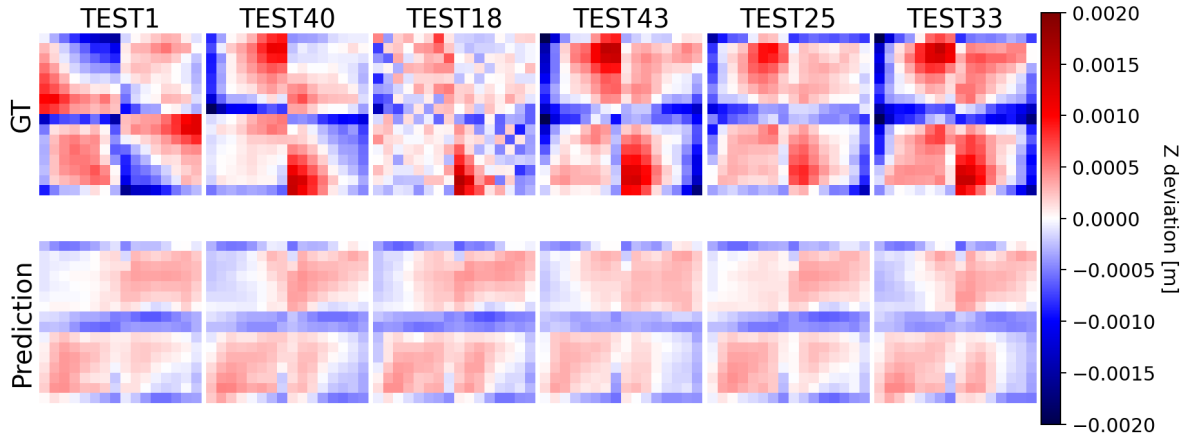
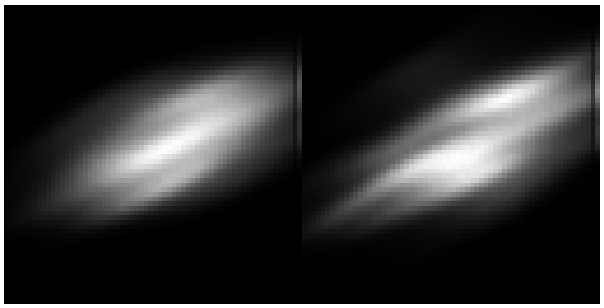
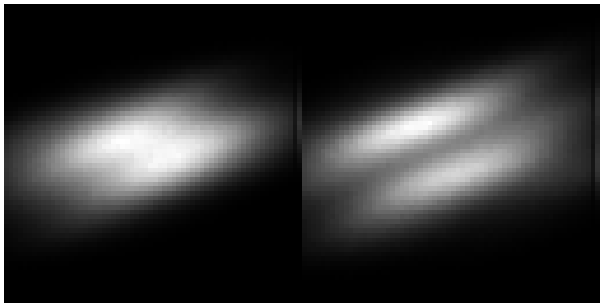


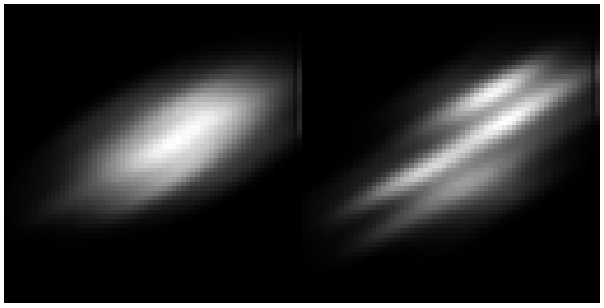
Figure A.37: Surface predictions



(a) Flux Density "TEST2"



(b) Flux Density "TEST20"



(c) Flux Density "TEST52"

Figure A.38: Comparison of flux density predictions (left) and ground truth (right)