Supplementary Material for PAINT: The First FAIR Database for Concentrating Solar Power Plants

Anonymous Authors

Keywords: Concentrating solar power plant, Operational data, FAIR data

Detailed Information on the Database

This section provides a comprehensive overview of the PAINT database, detailing the structure and content of the included data. It also serves as a user guide to facilitate effective navigation and interpretation of the database.

Tower Properties

The tower properties data is saved in the file WRI1030197-TOWER-MEASUREMENTS.JSON and contains information on the power plant and the two installed solar towers. Within this file the key *power_plant_properties* contains the global ID and coordinates of the power plant. Information on the three available calibration targets is available in the keys *solar_tower_juelich_upper*, *solar_tower_juelich_lower*, and *multi_focus_tower*. For each of these calibration targets information on the *type* (e.g. planar, convex cylinder), the *normal_vector* and the *coordinates* are provided. These coordinates define the borders of the calibration target (i.e. *upper_left*, *upper_right*, *lower_left*, *lower_right*) as well as the *center* (Figure 1). Finally, information on the *receiver* is provided including the same information on the type and normal vector and the center coordinate. For the receiver, both the inner and outer border values are provided (*receiver_outer_upper_left*, *receiver_outer_upper_right*, *receiver_outer_lower_right*, *receiver_outer_lower_left*, *receiver_inner_upper_left*, *receiver_inner_upper_right*, *receiver_inner_lower_right*, *receiver_inner_lower_left*) as shown in Figure 1.

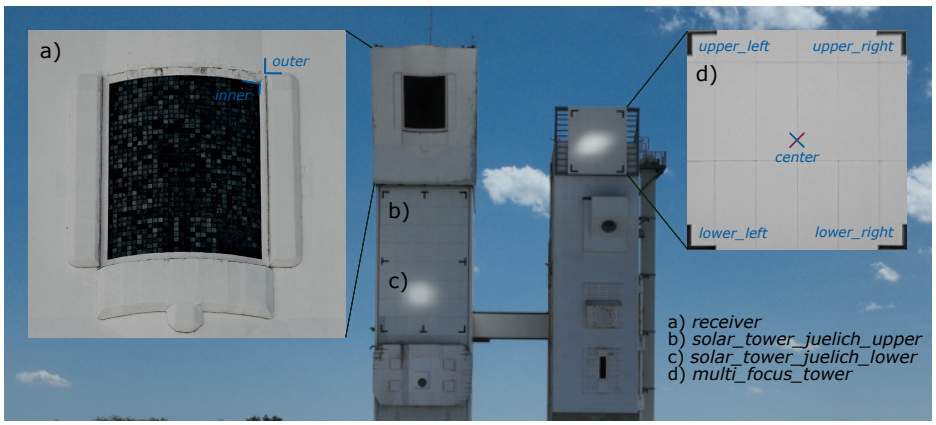

Fig. 1: **The tower coordinates.** An overview of the coordinates provided in the tower properties data. For the calibration targets only the outer borders and the center are provided, whilst for the receiver both inner and outer coordinates are supplied.

Table 1: **Available meteorological data.** Overview of weather variables from the two considered meteorological stations including the variable names, descriptions, units, temporal resolution, and data availability.

Variable Name	Description	Units	Temporal Resolution	Jülich	DWD
ATMOSPHERIC_PRESSURE	Atmospheric pressure.	hPa	1 s	✓	
CLOUD_COVER_1H	Total cloud cover.	fraction (1/8)	1 h		✓
DIFFUSE_IRRADIATION	Diffuse part of the solar irradiance.	$\mathrm{W\,m^{-2}}$	1 s	✓	
DIRECT_IRRADIATION	Direct part of the solar irradiance.	$\mathrm{W\,m^{-2}}$	1 s	✓	
GLOBAL_IRRADIATION	Total (diffuse + direct) solar irradiance.	$\mathrm{W\,m^{-2}}$	1 s	✓	
GLOBAL_RADIATION_10MIN	Sum of solar incoming radiation.	$\mathrm{J\,cm^{-2}}$	10 min		✓
HUMIDITY_1H	The humidity.	%	1 h		✓
LONG_WAVE_RADIATION_10MIN	Sum of longwave downward radiation.	$\mathrm{J\,cm^{-2}}$	10 min		✓
PRECIPITATION	Rainfall amount.	$\mathrm{mm\,d^{-1}}$	1 s	✓	
PRESSURE_VAPOR_1H	Vapor pressure.	hPa	1 h		✓
RELATIVE_HUMIDITY	Relative humidity.	%	1 s	✓	
SHORT_WAVE_RADIATION_10MIN	Diffuse solar radiation.	$\mathrm{J\,cm^{-2}}$	10 min		✓
SUNSHINE_DURATION_10MIN	Duration of sunshine.	h	10 min		✓
TEMPERATURE	Ambient air temperature.	°C	1 s	✓	
TEMPERATURE_DIFFUSE	Temperature related to diffuse irradiance.	°C	1 s	✓	
TEMPERATURE_DIRECT	Temperature related to direct irradiance.	°C	1 s	✓	
TEMPERATURE_GLOBAL	Temperature related to global irradiance.	°C	1 s	✓	
TIME	Timestamps.	-	1 s	✓	
VISIBILITY_RANGE_1H	Range of visibility.	m	1 h		✓
WEATHER_TYPE_1H	Encoded weather condition type.	-	1 h		✓
WIND_DIRECTION	Wind direction.	Degrees (0–360)	1 s	✓	
WIND_SPEED	Wind speed.	$\mathrm{m\,s^{-1}}$	1 s	✓	

Weather Data

Meteorological weather data is provided from a weather station at the Jülich tower and from the DWD weather station Aachen-Orsbach with the ID 1500. An overview of all variables available across both data sets is provided in table 1, and we describe the available data for each weather station in more detail in the following.

Jülich Tower

Meteorological data is recorded at a 1 s resolution with a station at the Jülich solar tower. Due to this high resolution, we group the data per month and save each months data into a separate HDF5 file, e.g. YYYY-MM-JUELICH-WEATHER.H5. Each HDF5 file is organized in a flat structure where all variables are stored as individual datasets at the root level of the file. Each dataset represents a distinct meteorological measurement and due to the constant 1 s resolution for all variables has the same length. Metadata attributes are embedded within each dataset, including a *description* that briefly defines the physical meaning of the variable, and *units* specifying its measurement units. Temporal information is stored in a separate *time* dataset, with all variables aligned to this time axis.

Please note that the Jülich weather data is presented without any pre-processing or guarantee of accuracy. Accurate performance of weather stations requires regular calibration, alignment, and maintenance. In the case of the Jülich weather station, such maintenance is only ensured during specific experimental periods. As a result, discrepancies in the data may occur. For a more reliable, though lower-resolution, alternative, we refer readers to the accompanying dataset provided by the DWD.

DWD Weather Station

We supplement the Jülich weather data with measurements from the Aachen-Orsbach DWD weather station. These measurements are either at a 10 min or 1 h resolution and saved in a single HDF5 - DWD-WEATHER.H5. This file is hierarchically organized by weather station identifier, to enable the integration of further weather stations at later date. Since we currently only consider one weather station, all data is stored under a single station group (*15000*), representing the ID of the Aachen-Orsbach station. Each meteorological variable is contained within its own subgroup, named according to the variable and its temporal resolution (e.g., *humidity_1h*, *global_radiation_10min*). These subgroups each include two datasets: *time*, containing the measurement timestamps, and *value*, containing the corresponding measurement data. This separation is necessary due to the different temporal resolutions contained within the dataset. We also embed descriptive metadata within each

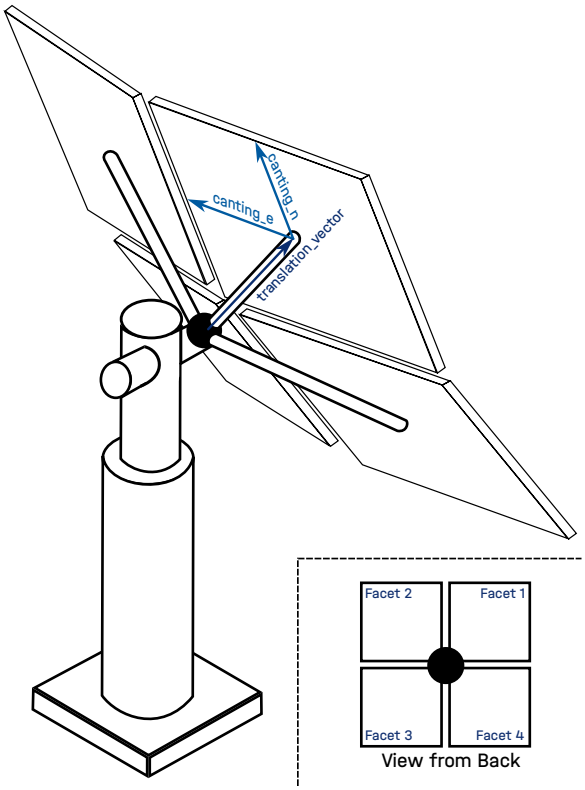

Fig. 2: **Heliostat Facet Properties.** An overview of the translation and canting vectors used to describe the facets for each heliostat.

variable group via *description* and *units* attributes. Additionally, the station group itself includes attributes such as *station_name*, *state*, *latitude*, *longitude*, and *height*.

Heliostat Properties

For each heliostat included in the database, we provide a dedicated JSON file containing comprehensive property information, named using the format: `[HeliostatID]-heliostat-properties.json`. This file includes essential metadata such as the heliostat's geographic position (*heliostat_position*, with latitude, longitude, and elevation), its physical dimensions (*height* and *width*), and any available information on past *renovation* activities. The file also specifies the *initial_orientation* vector, which indicates the default direction the heliostat is configured to face upon installation.

Beyond these core parameters, the properties file provides detailed descriptions of both the *facet_properties* and the *kinematic_properties* of the heliostat. The *facet_properties* section includes the *canting_type*, which defines the strategy used to align the mirror facets (e.g., receiver canting), as well as the total *number_of_facets*. Each facet is individually described with its own metadata, including a *translation_vector* that defines its position relative to the heliostat's center, and canting direction vectors - *canting_e* (east) and *canting_n* (north) - that characterize its specific orientation (see Figure 2).

Additionally, the *kinematic_properties* section provides detailed information about the heliostat's kinematic structure and actuator configuration. Each heliostat in the PAINT dataset uses the same type of kinematic, based on two actuators responsible for movements in different directions (See Figure 3). Since each joint responsible for rotation introduces a mechanical offset, it is important to describe these offsets explicitly (see Figure 3). To this end, the file includes translation vectors in the east, north, and up directions for three key components: joint one, joint two, and the concentrator. These are represented by the following nine parameters: *joint_translation_e_1*, *joint_translation_n_1*, *joint_translation_u_1*, *joint_translation_e_2*, *joint_translation_n_2*, *joint_translation_u_2*, *concentrator_translation_e*, *concentrator_translation_n*, and *concentrator_translation_u*. Furthermore, the file includes a comprehensive description of the *actuators* that drive movement at each joint. These parameters define the mechanical and control characteristics

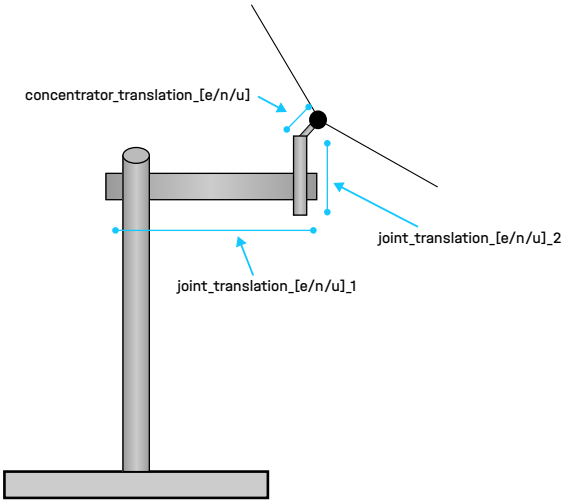

Fig. 3: **Kinematic Translations.** A side on view of the heliostat highlighting the joint and concentrator translation vectors used to describe the kinematic.

Table 2: An overview of the actuator parameters stored for each actuator. All heliostats in Jülich are equipped with two actuators.

Parameter Name	Description	Source	Reference
type_axis	Type of actuator motion (e.g., linear, rotary).	Measured	–
min_increment	Minimum step the actuator can move.	CAD	–
max_increment	Maximum actuator increment range.	Measured	–
increment	Total number of increments per full stroke.	CAD	–
offset_shift	Adjustment in the actuator's zero position.	CAD	(4) in Figure 4
initial_stroke_length	Initial extension length of the actuator.	HeliOS	(5) in Figure 4
offset	Physical offset from the actuator axis to the pivot.	CAD	(3) in Figure 4
pivot_radius	Radius from the pivot center to actuator anchor.	CAD	(2) in Figure 4
radius_shift	Shift in pivot radius due to geometry.	CAD	(1) in Figure 4
clockwise_axis_movement	Boolean indicating movement direction: 0 = counterclockwise, 1 = clockwise.	CAD	–
initial_angle	Starting angular position.	HeliOS	–
min_movement_angle	Minimum angular range allowed.	CAD	–
max_movement_angle	Maximum angular range allowed.	Measured	–
movement_speed	Speed at which actuator moves.	CAD	–

of each actuator and are essential for modeling and simulation. The actuator parameters are either taken from CAD models provided by the manufacturer, measured directly on the heliostat, or from an *HeliOS* optimization run during installation. A summary of the actuator parameters is provided in Table 2, while Figure 4 offers a visual representation to further clarify their configuration.

Calibration Data

Calibration data is recorded for 1893 heliostats and consists of calibration target images and calibration properties. In this section we explain these two data types in more detail.

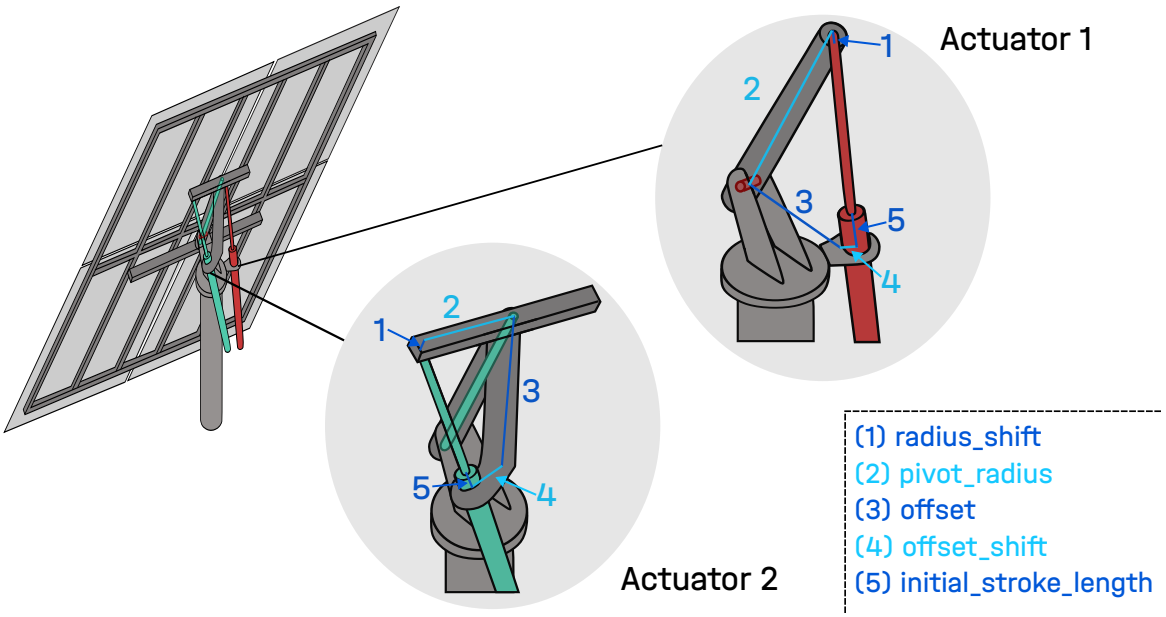

Fig. 4: **Actuator Parameters.** An illustration of the actuator parameters for each of the two actuators included on the heliostats in Jülich.

Calibration Properties

For every calibration image we save the associated properties in a JSON file prefixed by the calibration image ID - [CalibrationID]-calibration-properties.json. This file contains the field *motor position* which contains the motor position for axis one and axis two at the time the calibration image was recorded. The motors installed in the heliostats in Jülich are stepped motors, where axis one rotates vertically and axis two horizontally. Therefore these motor positions indicate the step increment of these motors at the time the calibration image was recorded. Additionally the file contains the *target_ name* which indicates which calibration target was used, and extracted *focal_ spot* centroids, using both the measurements from the *HeliOS* system and *UTIS* (see Technical Details on Applied Methods). Finally we record the *sun_ elevation* and *sun_ azimuth*. For the azimuth measurements we use the astronomical (south-based) azimuth system which is measured clockwise from due south, i.e. a azimuth of 0° implies the sun is due south whilst an absolute value of 180° implies the sun is due north [1].

The calibration properties file contains all the information necessary to evaluate a calibration algorithm. Therefore, if a user does not want to work with raw image data, they can choose to download only the calibration properties files and evaluate their calibration method exclusively based on this data. Not only does this simplify the workflow for users inexperienced with image analysis, but also drastically reduces the size of the data to be downloaded.

Calibration Images

We provide in total 218 713 calibration target images in the PAINT database. Additionally, we supply pre-processed versions of these calibration images to simplify the image processing workflow. The four variants of the image we provide are described below:

- **Raw Image:** The raw image saved in a PNG file with the name ID-raw.png, contains the original raw calibration target image captured during the measurement (see Technical Details on the Applied Methods). The raw image may contain multiple calibration targets, be slightly skewed, or taken from various positions. Therefore, in order to effectively work with these images processing is required.
- **Cropped Image:** The cropped image saved in a PNG file with the name ID-cropped.png, contains a cropped grayscale version of the original image, showing only the target relevant for the calibration measurement. This cropping is performed via target matching (see Technical Details on the Applied Methods).
- **Flux Image:** The flux image saved in a PNG file with the name ID-flux.png, is obtained with the UTIS model by decomposing the cropped target image into background illumination and a

relative flux distribution that accounts for the targets reflectivity (see Technical Details on the Applied Methods).

- **Flux Centered Image:** The centered flux image saved in a PNG file with the name ID-FLUX-CENTERED.PNG, contains a further processed version of the flux image where the relative flux distribution is now centered around the centroid of the focal spot. Additionally the image is scaled to a uniform size to simplify data processing (see Technical Details on the Applied Methods).

Deflectometry Data

We collect deflectometry data from 471 heliostats in the form of deflectometry measurements and a summary of results for each measurement. In total the PAINT database contains 654 deflectometry measurements, which we describe in more detail in the following.

Results Summary

For each deflectometry measurement the QDec_2014-101 measurement system automatically generates a summary of the results as a PDF file which we include in PAINT with the file name [HELIOSTATID]-[TIME]-DEFLECTOMETRY-RESULT.PDF. This PDF contains some descriptive information on the measurement, an overview of the results, and graphs of the results. The data from this PDF is not useful for power plant operations, however is included to provide the interested user with a complete overview of the conducted measurements.

Deflectometry Measurements

The deflectometry measurements are saved in an HDF file containing the heliostat ID and the time of the measurement in the name, i.e. [HELIOSTATID]-[TIME]-DEFLECTOMETRY.H5. The HDF5 is structured with individual fields for each facet, i.e. *facet1*, *facet2*, and so forth. In the PAINT database all heliostats are comprised of four facets and therefore each HDF5 file has four facet fields. Within each facet field the *surface_normals* and *surface_points* are saved. The surface points are a 3D point cloud representation of the surface, whilst the surface normals describe vectors perpendicular to the surface. During the original measurements, there are often missing values which means the number of points recorded per facet can vary noticeably. To fix this issue, the filled deflectometry measurements are provided in an additional HDF5 file - [HELIOSTATID]-FILLED-[TIME]-DEFLECTOMETRY.H5 - where missing values are replaced with ideal vectors. This ensures there are always 80 760 or 80 759 measurements per facet.[1] These filled deflectometry measurements are provided directly during the measurement campaign from the QDec_2014-101 measurement system. Therefore, these filled measurements can be considered as additional raw data from the measurement and not data that we have additionally processed.

STAC Specification & Metadata

The SpatioTemporal Asset Catalog (STAC) specification is a standardized framework for describing and discovering geospatial data assets across space and time. Originally developed for satellite imagery, STAC has evolved to support a wide array of spatiotemporal data sources, including aerial and drone imagery, hyperspectral data, synthetic aperture radar, point clouds, lidar, video, and derived products such as NDVI and mosaics. At its core, STAC defines a minimal set of JSON-based object types - Item, Catalog, and Collection - which are connected via hyperlink relations, enabling both simple static catalogs and dynamic, API-driven query interfaces.

STAC is designed to be modular and extensible, allowing for the addition of domain-specific metadata through standardized extensions. The Item object, which represents a single geospatial asset, is a GeoJSON Feature augmented with metadata fields to describe temporal coverage, licensing, associated assets, and more. The Catalog and Collection objects organize Items into navigable and discoverable structures. In the following we describe in more detail the fields contained within a STAC Catalog, Collection, and Item and how we apply this structure to the PAINT database.

STAC Catalog

The STAC Catalog serves as the top-level container that enables structured navigation and discovery of data within a dataset. It is designed to support simple and consistent metadata linking, allowing

[1] The number of filled points is constant for all facets on a given heliostat but varies accross heliostats. The number of points filled is determined by the QDec_2014-101 measurement system.

both humans and machines to traverse complex data hierarchies. A STAC Catalog includes core fields such as *type*, which is always set to `Catalog` for a STAC catalog, *id*, *title*, and *description*. In addition, it contains an array of *links* that define relationships to other STAC entities. These typically include a *self* link pointing to the catalog itself, a *root* link referencing the top-most catalog in the hierarchy, and optional *child* links to nested catalogs, collections, or items.[2]

In PAINT, the full dataset is organized under a root catalog, `WRI1030197-catalog-stac.json`, which serves as the primary entry point. This catalog links to all major components of the dataset, including per-heliostat data, weather observations, and tower measurements. Each individual heliostat is associated with its own catalog (e.g., `[HeliostatID]-catalog-stac.json`), which in turn links to collections representing specific types of data acquired for that heliostat. These nested catalogs support a modular and extensible structure, as further detailed in the following section on STAC Collections.

STAC Collection

A STAC Collection extends the base Catalog structure by providing additional descriptive fields, including *extent* (both spatial and temporal coverage), *license*, *keywords*, and *providers*, which facilitate improved searchability and discovery across datasets. As with catalogs, collections also include a set of *links* that define their position in the overall hierarchy, such as references to their parent catalog or contained items. Collections are especially useful for organizing datasets with consistent acquisition methods or shared semantic meaning.[3]

In PAINT, we use STAC Collections to organize the different types of data available for each heliostat. Specifically, each heliostat catalog links to up to three data collections (depending on data availability): a deflectometry collection, a properties collection, and a calibration collection. This design ensures that semantically distinct datasets are kept separate while still being easily discoverable and linked through a shared catalog structure. Additionally, the weather data collected from both the Jülich tower and the DWD weather station is organized as a separate collection, allowing users to access temporal and spatial metadata relevant to environmental conditions.

STAC Item

A STAC Item represents a single spatiotemporal asset, typically corresponding to a specific observation, measurement, or data file. It is the atomic unit in the STAC model and must include a unique *id*, a *geometry* and *bbox* describing its spatial footprint, a *datetime* or temporal range, and a set of *properties* that capture additional metadata. Crucially, each item includes one or more *assets*, which are pointers to the actual data files, along with descriptive metadata such as MIME type and title. As with Catalogs and Collections, Items are linked to their parent collections or catalogs.[4]

In PAINT, STAC Items are used extensively to represent individual measurements across multiple data types. Within the calibration and deflectometry Collections, separate items are created for each deflectometry and calibration measurement. Additionally, each heliostat includes a single item describing its calibration properties in the calibration collection. These items may contain multiple assets, e.g. both raw and pre-processed versions of deflectometry and calibration data, allowing users to select the most appropriate data format for their needs. For weather data, items represent either the entire set of the DWD data or monthly subsets of Jülich station data. A single item is also used to represent tower properties, placed directly within the root catalog.

STAC Extensions

A key feature of the STAC specification is its extensibility. While the core specification defines a minimal set of fields, extensions introduce additional metadata fields that are collaboratively developed within the community to ensure broad applicability and interoperability. STAC maintains a centralized overview of available extensions, including their maturity and ownership to support reuse and transparency.

We make use of two extensions in our STAC metadata, the VIEW extension (https://github.com/stac-extensions/view) and the PROCESSING extensions (https://github.com/stac-extensions/

[2] The full STAC Catalog specification is available at: https://github.com/radiantearth/stac-spec/blob/master/catalog-spec/catalog-spec.md

[3] The full STAC Collection specification is available at: https://github.com/radiantearth/stac-spec/blob/master/collection-spec/collection-spec.md

[4] The full STAC Item specification is available at: https://github.com/radiantearth/stac-spec/blob/master/item-spec/item-spec.md

processing). From the VIEW extension we use the fields *view:sun_azimuth* and *view:sun_elevation* within the calibration item *properties* field to record the sun azimuth and sun elevation at the time the calibration image was recorded. From the PROCESSING extension we use the fields *processing:lineage* and *processing:software* within the description of various assets in a calibration item, to describe the processing performed to crop the calibration image, center it, and extract the centroid of the flux density.

Technical Details on Example Uses of PAINT Data

This section provides detailed implementation and data processing information for the example use cases presented in the main paper. We provide complete code to replicate these results as well as instructions on how to execute the code via GitHub.

Focal Spot Centroid Detection & Calibration

We demonstrate the benefit of the calibration data by performing heliostat calibration with PAINT data in ARTIST. In a first step, we download the calibration metadata using the STAC client provided in our software. Based on this metadata we then download calibration data for all heliostats that contain enough calibration measurements, i.e. we set a threshold for the minimum number of calibration measurements per heliostat and only consider heliostats that exceed this threshold. The final step of data pre-processing, is to check the calibration properties for each measurement to ensure a valid focal spot for both HeliOS and UTIS is provided. Since UTIS is a vision based approach for focal spot extraction [2], in a small number of cases the focal spot centroid extraction failed and no measurement is available. These measurements are then discarded.

To perform the calibration, we generate an ARTIST scenario based on the collected list of heliostats, and their associated properties. We then use the `KinematicReconstructor` in ARTIST to perform the calibration. In this step, kinematic deviations are fitted in ARTIST to ensure an accurate kinematic model is present. Internally this process involves ARTIST using differentiable ray tracing to predict an irradiance and extract the focal spot centroid. This predicted centroid is then compared to the measured centroid, and the resulting loss is minimized using a gradient-based Adam optimization to fit a geometric kinematic model. Once this gradient-based optimization has converged, the heliostat is considered calibrated and the final pointing area is computed. This calibration process is performed for both the UTIS and HeliOS centroids.

The pointing error per heliostat and per centroid extraction method are saved, which allows us to accurately compare the performance of the two different centroid extraction methods. Furthermore, since we use a full digital twin, the heliostat position is also tracked and available, therefore we can analyze the calibration error as a function of the distance from the tower.

Solar Flux Prediction and Heliostat Mirror Characterization

The show the importance of the deflectometry data in the PAINT database we also use ARTIST to generate solar flux predictions with and without this deflectometry data. For this purpose, we select three heliostats and one calibration measurement for each heliostat, specifically AA39 with calibration measurement 149576, AY26 with calibration measurement 247613 and BC34 with calibration measurement 82084. Each of these heliostats also has deflectometry measurements that are available.

In the first step, we create two ARTIST scenarios. In the first scenario we do not use any information from the deflectometry file, i.e. we load a scenario with the three heliostats above and only use the information on the facet translation and canting to create an idealized surface, i.e. without any deformations. In the first scenario, we do use the deflectometry measurements. Specifically, we fit a NURBS surface using the surface normals from the deflectometry measurement resulting in a scenario containing the same three heliostats, but taking account of the deformations in their surfaces.

We can now use these two scenarios to generate solar flux predictions with ray tracing in ARTIST. We aim to replicate the flux image from the calibration measurements mentioned above by considering the same incident ray direction and assuming a simplified model of the sun. We then use the `HeliostatRayTracer` in ARTIST to perform ray tracing and generate the prediction. We compare these images to the flux images extracted via UTIS for the same calibration measurement. Finally, we visualize the surface normals from a deflectometry measurement to quantify the surface deformation, i.e. the deviation from a perfectly flat surface.

Technical Details on the Standardized Calibration Benchmarks Derived from PAINT

This section outlines the technical methodology used to generate standardized calibration benchmark datasets from PAINT data. Each method aims to produce distinct training, validation, and test sets that reflect different operational scenarios of solar tower power plants. All splits are constructed based on solar position metadata, such as azimuth, elevation, and elliptical longitude, to ensure that benchmarks reflect realistic and challenging calibration conditions. Furthermore, if a heliostat does not contain enough data to fulfill the requirements (i.e. specified training or validation size) then this heliostat is removed from the benchmark dataset.

Azimuth Split

The *Azimuth Split* method divides the dataset based on the sun's azimuth angle. First, all calibration data samples are sorted by their associated azimuth value. Since we use the solar azimuth measured from due south, a large azimuth value indicates the sun is low on the horizon whilst a low azimuth value indicates the sun in near-zenith, i.e. the highest evaluation for the day. The splits are then determined to ensure high variation between the training and validation splits. Specifically, the samples with the lowest azimuth values are used for training whilst those with the highest azimuth values are used for validation. All remaining samples are used for testing. When creating benchmark datasets using the azimuth method, the training and validation size are specified, with the test size per heliostat varying based on how much data remains.

High-Variance Split

The *High-Variance Split* method constructs the dataset splits using a distance-based diversity metric applied to the solar azimuth and elevation angles. For each calibration data point, a feature vector is constructed using its azimuth and elevation. A pairwise distance matrix is then computed between all data points, using the k-nearest neighbors distance in Euler coordinates. The validation dataset is formed by iteratively selecting the data point that has the highest minimum distance to all previously selected points, thereby maximizing diversity. Once the validation set is complete, the same process is applied to the remaining data to form the test set. The training set is then constructed from the remaining samples using the same selection strategy until the desired dataset size is reached. This approach ensures maximal dissimilarity between training, and validation sets, creating a benchmark that evaluates a method's ability to generalize across widely varying sun positions. When creating benchmark datasets using the high-variance method, the training and validation size are specified, and the test size is equal to the validation size.

Balanced Split

The *Balanced Split* method uses clustering to ensure that all three datasets cover a representative distribution of sun positions. First, a feature space is constructed using the azimuth and elevation of the sun for each calibration sample. The entire dataset is then clustered using the kMeans algorithm, where the number of clusters equals the intended size of the validation dataset. From each cluster, one sample is randomly selected to form the validation dataset. A second sample is selected, if available, to form the test dataset. If a cluster contains fewer than two samples, additional samples for the test set are drawn randomly from the remaining data to maintain balance. The remaining data is used to form the training set via random sampling. This method ensures that the training, validation, and test sets are all evenly distributed in solar position space, which supports evaluation under typical and average conditions across the sun's path. When creating benchmark datasets using the balanced method, the training and validation size are specified, and the test size is equal to the validation size.

Solstice Split

The *Solstice Split* method generates seasonally distinct datasets by using the elliptical longitude of the sun. Each data point is annotated with its corresponding elliptical longitude, computed from the time and location of the observation. The data is sorted by elliptical longitude, and the portion of data closest to the winter solstice is selected for the training set. The portion closest to the summer solstice is selected for the validation set. All remaining samples are used for testing. When creating benchmark datasets using the solstice method, the training and validation size are specified, with the test size per heliostat varying based on how much data remains.

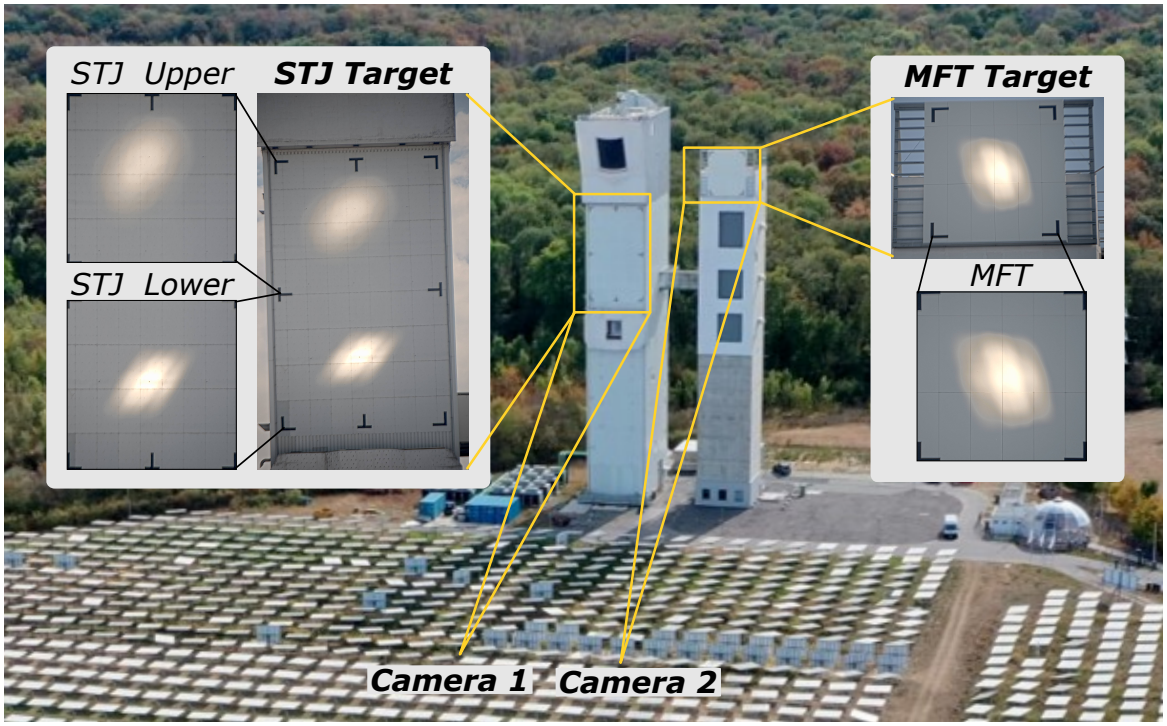

Fig. 5: **Calibration Measurement Setup.** Measurement setup of the heliostat calibration at the solar tower in Jülich showing the multiple calibration targets and the location of the cameras.

Technical Details on the Applied Methods

In this section we present technical details on our applied methods, first considering the measurement techniques and afterwards the pre-processing methods applied to the dataset.

Measurement Methods

Whilst heliostat properties data was collected via simple laser measurements or taken from the manufactures specifications, the calibration and deflectometry data were obtained via measurement campaigns. We explain the setup and methods used to obtain these measurements in detail in the following.

Heliostat Calibration

Heliostat calibration at the Jülich facility is performed using the Camera-Target Method, commonly known as the Stone method [3]. This approach redirects the heliostat's focal spot from its intended receiver onto a Lambertian white calibration target placed near the receiver structure. A camera system captures images of the reflected focal spot on the target, enabling analysis of the heliostat's optical performance. The Jülich facility includes three distinct calibration targets: the Solar Tower Jülich Upper, the Solar Tower Jülich Lower, and the Multi Focus Tower target as shown in Figure 5. Alongside the captured images, the system records the heliostat's motor positions - describing its kinematic configuration - as well as the solar position at the time of measurement, including azimuth and elevation angles. This combination of image data, actuator states, and solar geometry constitutes the raw calibration data. The images are subsequently pre-processed,including cropping and the identification of the focal spot centroid, as detailed in the Pre-Processing Methods section.

Deflectometry Measurement

Deflectometry is an optical measurement technique used to characterize the surface shape and quality heliostats. It operates by projecting a known light pattern—typically a fringe or grid—onto the reflective surface and capturing the reflected image using a calibrated camera system. Deviations in the reflected pattern are analyzed to infer surface slope and curvature, allowing for precise reconstruction of the mirror's surface geometry. This non-contact method is highly sensitive and capable of detecting small deformations such as warping, waviness, or other optical defects. In the context

of solar tower power plants, deflectometry has become the most widely recommended method for assessing the optical performance of heliostat mirrors [4]. The deflectometry measurements included in the PAINT database were performed with the `QDec_2014-101` software from CSP Services GmbH and deailed information on the algorithms used to derive the measured surface points and surface normals from the images is not publicly available.[5].

Despite its demonstrated effectiveness, deflectometry measurements remain underutilized in commercial settings due to challenges in automation, slow measurement throughput, and high operational and maintenance costs [5]. As a result, mirror surface quality is often neglected in day-to-day plant operation, making systematically collected deflectometry data—such as those available via PAINT valuable for research and quality assurance in solar thermal applications.

Pre-Processing Methods

Prior to inclusion in the PAINT database, the raw data collected at the Jülich solar tower underwent a series of pre-processing steps. These included organizing unstructured datasets, unifying data formats, performing coordinate transformations, and applying image processing techniques to extract relevant features.

Data Organization and Format Standardization

The original data were sourced from various unstructured dumps, distributed across inconsistent directory hierarchies. A critical early step was to reorganize these datasets into a standardized structure aligned with the PAINT database schema. For example, calibration images were linked to specific heliostats based on a mapping provided in Excel spreadsheets; the images were then relocated into folders organized by heliostat ID.

Deflectometry measurements were originally stored in a proprietary binary format. These were converted to HDF5 to ensure consistency across the database. The heliostat ID, extracted during conversion, was also used to place the data into the appropriate directory structure, mirroring the organization used for calibration images.

Heliostat property data were compiled from multiple heterogeneous sources, including CSV and Excel files detailing locations, facet configurations, and kinematic parameters. These were merged into unified, JSON-based property files, each stored according to heliostat ID.

Weather data from the Jülich tower were initially provided as raw text files. These were parsed and converted into HDF5 format. To facilitate efficient access and download, the data were partitioned into separate HDF5 files, each representing a single month.

Coordinate Conversion

An essential component of the pre-processing pipeline involved transforming spatial coordinates from the Gauss-Krüger (GK) coordinate system to the globally recognized WGS84 reference system. Much of the original data from the Jülich solar tower was recorded in GK zone 2 (EPSG:31466), a projected coordinate system commonly used in Germany for engineering and topographic applications. Unlike WGS84, which represents geographic positions using latitude, longitude, and optionally elevation, the GK system is based on the Bessel ellipsoid and uses a transverse Mercator projection, expressing positions in metric units (easting and northing) within defined 3-degree longitudinal zones.

While the GK system offers high local precision, it lacks interoperability with global geospatial standards such as those used in the STAC specification. Therefore, consistent and accurate coordinate transformation was necessary across all spatial metadata. This transformation was implemented using the `pyproj` library, which performs both projection and datum conversions. A `Transformer` object was instantiated with EPSG:31466 as the source and EPSG:4326 (WGS84) as the target. This transformation was applied uniformly across all metadata referencing spatial locations — including heliostat positions, calibration targets, and tower coordinates - to ensure geospatial consistency throughout the dataset.

Image Processing and Feature Extraction

To facilitate the use of calibration data and support algorithm development without requiring raw image processing, we also included pre-processed versions of the calibration images, along with extracted features. Image cropping was performed using a template-matching algorithm based on

[5] https://www.cspservices.de

known markers on the calibration targets. Once cropped, further processing was conducted using a pre-trained deep learning model, UNet-Based Target Image Segmentation (UTIS), available at https://github.com/DLR-SF/UTIS-HeliostatBeamCharacterization/tree/main. UTIS was used to extract the focal spot by learning spatial features from the images. The resulting pre-processed images — converted to grayscale and centered on the focal spot — as well as the extracted focal spot centroids were stored in the database. A detailed description of the UTIS model and its training methodology is provided in [2]. Alternatively, we also include the focal spot measured via the *HeliOS* system during the calibration process.

STAC File Generation

In addition to the organizational and analytical steps described above, another central task was the generation of STAC-compliant metadata files. This process was conducted in parallel with data conversion and feature extraction. Relevant metadata were aggregated from various sources and structured into the STAC specification using the standard hierarchy of Catalogs, Collections, and Items. The generated STAC files are available as part of the PAINT package and the schema used for generating these STAC files is available via GitHub.

References

[1] Duffie, J. A., Beckman, W. A. & Blair, N. *Solar engineering of thermal processes, photovoltaics and wind* (John Wiley & Sons, 2020).

[2] Kuhl, M. *et al.* In-situ unet-based heliostat beam characterization method for precise flux calculation using the camera-target method. *Solar Energy* **279**, 112811 (2024).

[3] Stone, K. W. Automatic heliostat track alignment method (1986).

[4] März, T., Prahl, C., Ulmer, S., Wilbert, S. & Weber, C. Validation of two optical measurement methods for the qualification of the shape accuracy of mirror panels for concentrating solar systems. *J. Sol. Energy Eng.* (2011).

[5] Ulmer, S., März, T., Prahl, C., Reinalter, W. & Belhomme, B. Automated high resolution measurement of heliostat slope errors. *Solar Energy* **85**, 681–687 (2011).