Electronic Communications of the EASST
Volume 83 Year 2025

deRSE24 - Selected Contributions of the 4th Conference for Research Software Engineering in Germany

Edited by: Jan Bernoth, Florian Goth, Anna-Lena Lamprecht and Jan Linxweiler

Documenting ML Experiments in HELIPORT

David Pape, Oliver Knodel, Sebastian Starke

DOI: 10.14279/eceasst.v83.2597

License: 🅭 🅯 This article is licensed under a CC-BY 4.0 License.

Electronic Communications of the EASST (https://eceasst.org).
Published by **Berlin Universities Publishing**
(https://www.berlin-universities-publishing.de/)

Documenting ML Experiments in HELIPORT

David Pape[1], **Oliver Knodel**[2], **Sebastian Starke**[3]

[1] d.pape@hzdr.de, https://orcid.org/0000-0002-3145-9880
[2] o.knodel@hzdr.de, https://orcid.org/0000-0001-8174-7795
[3] s.starke@hzdr.de, https://orcid.org/0000-0001-5007-1868
Department of Information Services and Computing
Helmholtz-Zentrum Dresden – Rossendorf, Germany

Abstract: Machine learning practitioners use a variety of tools to track their experiments. These tools have in common that they are only concerned with the machine learning aspect of the experiment: They may track model parameters or performance metrics of the model, but provenance of the training data or scientific outcomes produced with the trained model are largely overlooked. This is a drawback especially when it comes to experiments where machine learning meets scientific experiments and traditional simulations. In this contribution we present an initial evaluation on improving documentation of such machine learning experiments using our data management guidance system HELIPORT. We also explore existing experiment tracking tools and metadata schemas for ML experiments in the process and discuss their suitability for integration with HELIPORT.

Keywords: data management, research software engineering, machine learning, metadata, ontologies

1 Introduction

HELIPORT [KVU+20, VUS+23] is a data management guidance system that aims at making the components and steps of the entire research experiment's life cycle findable, accessible, interoperable and reusable according to the FAIR principles. It integrates documentation, computational workflows, data sets, the final publication of the research results, and many more resources. This is achieved by gathering metadata from established tools and platforms and passing along relevant information to the next step in the experiment's life cycle. HELIPORT's high-level overview of the project allows researchers to keep all aspects of their experiment in mind.

A particularly interesting future use case for HELIPORT could be the documentation of research which, in addition to physical experiments, involves machine learning (ML). At Helmholtz-Zentrum Dresden - Rossendorf (HZDR), the local Helmholtz AI[1] unit—consisting of a team of consultants and a young investigator group—is involved in many of these experiments.

ML experiments in these settings are often prototypical in nature and driven by iterative development, so reproducibility and transparency are a great concern. It is essential to keep track of the relationship between input data, choices in model parameters, the code version in use, as well as performance measures and generated outputs at all times. This requires a data management platform that automatically records the changes made and their effects. Moreover, results

[1] https://www.helmholtz.ai

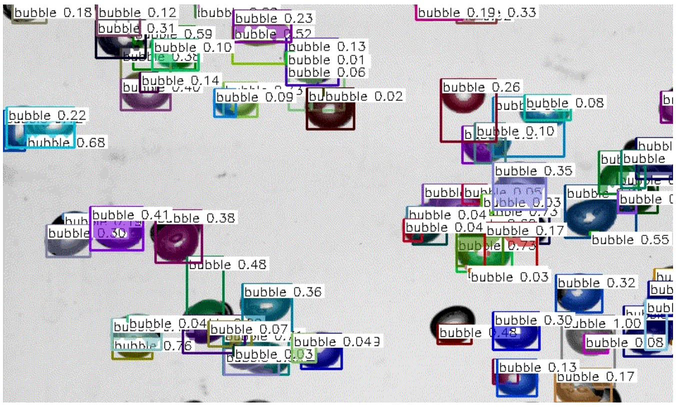

Figure 1: Instance segmentation using ML. Convolutional neural networks are used to detect bubbles in images of air-water flows [HSA+22a]. Shown here is the application of a trained network [HSA+22b] on synthetic test data. Image: Sebastian Starke.

need to be put into relation to the real experiment. However, existing experiment tracking tools that practitioners actually use, such as *Weights & Biases* [Bie20] and *MLFlow* [ZCD+18], live entirely in the ML domain. Their workflow begins with the assumption that data is available and ends after model training or inference.

Our envisioned platform inter-operates with the domain specific tools already used by the scientists, and is able to extract relevant metadata. It can also make persistent any additional information such as papers the work was based on, documentation of software components, workflows, or failure cases, and make it possible to publish these metadata in machine-readable formats. Moreover, the platform should enable the comprehensible development of ML models alongside the experiment. This would allow different teams (e.g. experimentalists and AI specialists) to work together on the same project in a seamless manner, and help generate FAIRer outcomes. In the long term it should aid in establishing digital twins of facilities, and making their maintenance a part of the data management process.

In this paper, we will examine whether our data management system HELIPORT can become such a platform. We start by establishing use cases and stating our analysis approach in Section 2. In Section 3, we analyse the requirements for our use cases and discuss metadata formats and tooling, as well as features in HELIPORT which can be applied to the use cases. Finally, in Section 4, we discuss the results and provide an outlook for future work.

2 Approach

As the basis for our investigation, we selected a number of use cases that members of our local Helmholtz AI unit have worked on, and that cover the research areas of HZDR well.

Figure 1 exemplifies a use case from the research area Energy where ML is used in the processing of videos recorded during an experiment to detect gas bubbles in a liquid. While the example shown in the figure uses images that were artificially created by simulation software, the approach is applicable to many experiments in fluid dynamics that involve bubbly flows. In

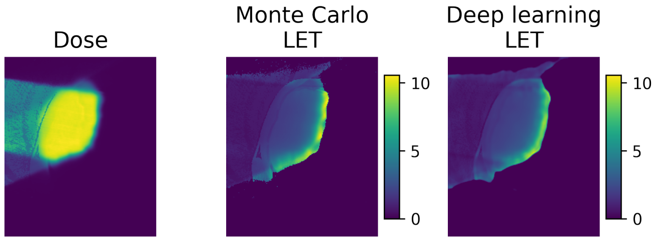

Figure 2: Convolutional neural networks are used as surrogate models for computationally expensive Monte Carlo simulations. The model predicts the linear energy transfer (LET) a patient is exposed to during proton-beam radiotherapy based on the three-dimensional dose distribution obtained during the treatment planning process. [SEZ+22] Image: Sebastian Starke.

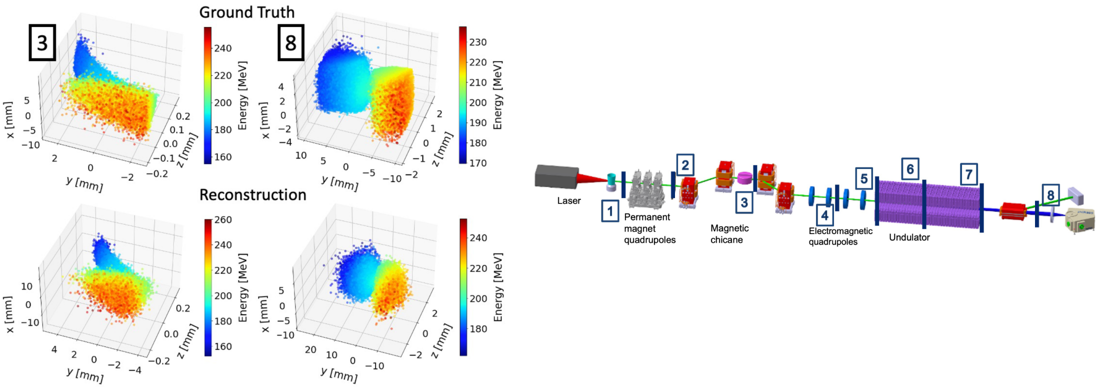

Figure 3: This example shows reconstructions of the phase space distribution at different locations along the COXINEL beamline [CAB+20] using normalizing flows neural networks. Images: Anna Willmann; Couprie et al.

Figure 2, an example from the Health research area is shown. Here, deep learning is used as an alternative to a more computationally expensive Monte Carlo simulation which determines the LET a patient is exposed to during radiation treatment. The COXINEL beamline shown in Figure 3 is an experiment from the Matter research area where surrogate models and virtual diagnostics are developed as part of a digital twin of the beamline.

We examined these use cases, identifying their requirements to the documentation system, and the metadata they produce which could be used for a more FAIR description of the experiment. We captured both the broader picture how documentation of these experiments could be improved, and collected concrete types of metadata that would be useful to document.

Next, we reviewed a number of tools for administering and tracking ML experiments that researchers use in their daily work. A variety of tools in this field exists. However, they typically lack the ability to collect provenance metadata of the used and produced artifacts [SS23]. We focused on the metadata they store about the ML experiment, and whether it can be extracted and used for documentation purposes. The particular tools looked at were TensorBoard, Weights & Biases (WandB), and MLflow, as these are popular options that were brought up in discussion with ML practitioners.

We also explored the landscape of ML metadata formats, specifically vocabularies and ontologies, focusing on the aspects of ML they cover (software, provenance, techniques, . . .) and whether established tooling exists that produces metadata in the given format, or makes use of them. Additionally, we presented our findings to colleagues from the ML field and fellow research software engineers (RSEs) to gather their feedback on the popularity of the metadata formats.

Finally, we assessed to what extent existing features in HELIPORT could be used to support the presented use cases and which additional features would need to be implemented.

3 Evaluation

The evaluation begins with the analysis of the requirements of the use cases with regard to documentation, and the metadata which can be collected for a more comprehensible description of the experiment. Afterwards, we examine different experiment tracking tools used by ML practitioners, as well as metadata formats, related tooling, and their popularity. Finally, we discuss features already available in HELIPORT and possible future improvements that could be used for documentation of the use cases.

3.1 Analysis of Use Case Metadata and Documentation Requirements

For all endeavors in ML, regardless of the use case, reproducibility is of key importance. Therefore, certain pieces of information about the training process need to be well documented:

Code and configuration The software, programs and scripts written for the purpose of the experiment and the used configurations.

Environment Software and runtime environment such as third-party tools and libraries used as dependencies, and container images.

Datasets and models Versioned training datasets, as well as trained models and their performance metrics (e.g. accuracy, precision) for each training run.

A shift in consciousness over the last years brought attention to energy consumption of computing efforts. To meet demands in this regard, the following should also be documented:

Compute resources used This comprises metrics such as the processing units (number of CPU and GPU cores), runtime (wall time), compute (e.g. in GPU hours) and used energy, both for training and inference.

To be able to describe machine learning as part of a larger experiments, the following requirements were identified:

Model provenance For each model, the software and training data used for its creation and their exact versions are known and can be retraced.

Project description across domains E.g. model inference can be described as a machine learning process but also as a computational workflow; data can be described in its role as training data but also as a published dataset with authorship information.

Identification of upstream changes Datasets used for training in ML experiments are often measurements from the real experiments and thus are subject to change. Changes in the dataset must be identifiable as such because they influence the model training and will cause different outcomes.

Identification of downstream benefits Users should be able to find the model and its outputs that facilitated results or findings, or that were used as a basis for decision making.

Collection of resources General collection of resources that are relevant for a project such as documentation, proposals, related projects, publications the work is based on (literature/reference management), new publications being worked on, training materials, tutorials, . . .

In order to not interfere with personal workflows of the researchers, the documentation tool should require little to no additional care when conducting experiments. Therefore, it should integrate seamlessly with their established personal workflows. Ideally, it could even help automate previously manual tasks.

Seamless integration with established tools, libraries, and workflows, such as MLflow, TensorBoard, Weights & Biases, Jupyter Notebooks, Python and shell scripts, . . .

Automation of manual tasks such as creation of assets for publications, e.g. visualization of any of the metadata mentioned above, textual or tabular representations such as basic model cards [MWZ+19].

MLOps for ML Experiments? A question that was raised in discussion with our colleagues was: “Can we support scientists by integrating MLOps in HELIPORT?” However, we found that MLOps and scientists performing ML experiments have quite different requirements. MLOps engineers are concerned with the administration of machine learning environments, including all the steps from the collection of datasets, training of the models, and deployment and monitoring of services in production. They aim for reliability of the infrastructure and reproducibility of the models. While ML scientists share the goal of reproducibility, they usually don’t administer any infrastructure or services. Models also might not even reach production status if they don’t produce the desired results. Moreover, for models that are deemed fit for their purpose, scientists often choose varying modes of interaction. They are rarely offered as part of an integrated service but usually shared on network or USB drives. While MLOps might become more relevant for such cases when domain scientists want to apply a developed model productively, we consider the documentation of model development and its relation to the experiment more important at the current point in time.

3.2 ML Experiment Tracking Tools

As mentioned above, the tools we assessed were chosen based on conversations with practitioners and their perceived popularity of the tools among their peers. MLflow and Weights & Biases (WandB) are both web services. While TensorBoard is also served as a website, it is not intended to be run as infrastructure; it is typically run locally or as part of a Jupyter Notebook. TensorBoard and MLflow are open source software, whereas WandB is a proprietary, commercially available software.

TensorBoard TensorBoard[2] is part of the TensorFlow ecosystem. Its makers describe it as a “visualization toolkit” as by itself, it does not provide extended tracking capabilities that other tools offer. When used with TensorFlow, it is usually loaded and launched from within a Jupyter notebook via a “magic command”. However, it is also possible to use it with other ML frameworks such as PyTorch[3] which provides a `SummaryWriter` class that can be used to write compatible metadata to disk. TensorBoard tracks and visualizes model performance metrics during the training process. It can also visualize operations and layers of a model in a graph, and display datasets (e.g. text or images). Since TensorBoard is a web application, all metadata can in theory be retrieved via its web API, however this is not intended as a general purpose interface. Provenance data is not recorded. For this purpose, the ML Metadata (MLMD) library[4] can be used. However, this requires more additional instrumentation of the code and the recorded metadata can not be displayed with TensorBoard. MLMD allows users to register datasets and models as artifacts, and record run code as executions. Artifacts and executions are connected via contexts which may represent, e.g., projects or experiments. Recorded metadata is stored in a database and can be queried, though it is only available in the library’s own data model, not in a standardized, interoperable format.

[2] https://www.tensorflow.org/tensorboard

[3] https://pytorch.org/

[4] https://www.tensorflow.org/tfx/guide/mlmd

Weights & Biases Weights & Biases[5] is an ML experiment tracking platform which can be used from a Python client library. Experiments are captured as “runs” which represent a training run with a given “config” (i.e., a set of hyperparameters, the dataset, independent variables). Runs can be tagged with arbitrary labels, e.g. to identify baseline or production models. Moreover, they can be provided with plain text notes, and grouped into namespaces called “projects”. The tracking of metrics is carried out via manual instrumentation of the user’s code. Calls to the log function take a dictionary of arbitrary metrics (e.g. accuracy, loss) or media (e.g. images, tables, plots), and the training step as their input. In addition, each logging call can also automatically track the state of the Git repository of the code, and system metrics such as GPU utilization. WandB can also track artifacts (e.g. datasets or models) which work similarly to files and directories. A model registry is provided which, among other features, can show “lineage maps” of the registered models, e.g. to visualize how models were produced in training runs and later used in evaluations. WandB allows users to create reports of their findings using formatted text and rich media including the created plots. These reports can be viewed via the website and exported as PDF or zipped LaTeX files. An export API for most of the recorded metadata is available, but export in a standardized format is not possible.

MLflow MLflow[6] allows users to track their ML experiments via Python, R, Java, and REST APIs. This means, user code has to be adapted to use the API. MLflow’s tracking capabilities are based on runs, i.e. executions of code, which can be grouped into experiments. For each run, model parameters, metrics and output files are recorded. Additional metadata can be tracked via tags. A predefined set of “system tags” is automatically tracked and includes, among other things, the Git repository and commit of the source code, and the Docker container’s image ID. Moreover, MLflow provides a model registry which allows registration, versioning, and annotation of model artifacts. While all recorded metadata can be retrieved via the API, it is not possible to extract any information in an interoperable format. This problem is tackled by MLflow2PROV [SS23] which we discuss in Subsection 3.4.

3.3 ML Metadata Formats

For this section, we focused our analysis on general machine learning ontologies and approaches that direct efforts at the documentation of workflows including Jupyter Notebooks. In sub-fields or related fields like data mining, a variety of ontologies such as Exposé [VS10], OntoDM-core [PSD14], and DMOP [KŁd+15] have been proposed. As these are usually only partially applicable to general ML, they were not considered. High-level ML ontologies often suffer from low adoption [SS23] but they will likely serve the high-level documentation approach of HELIPORT well.

MEX MEX [EMN+15] is a vocabulary for exchanging basic information about ML experiments, independent of their concrete implementation. It is based on the PROV ontology[7], reusing

[5] https://wandb.ai/site

[6] https://mlflow.org/

[7] https://www.w3.org/TR/prov-o/

its concept of entities, agents, and activities, and also uses terms from Dublin Core[8] and DOAP[9]. The vocabulary is split into three namespaces. MEX-Core comprises, among others, terms to describe experiments, their configurations and executions, as well as models and datasets. It can also describe the used hardware environment. MEX-Algorithm can be used to describe characteristics of ML algorithms such as their learning method (e.g. reinforcement or supervised), the learning problem (e.g. meta-heuristic or association), and the algorithm class (e.g. artificial neural network). MEX-Performance provides various measures, including classification, regression, and statistical measures, as well as the possibility of user-defined performance metrics. Through its high-level approach MEX aims to achieve high interoperability.

ML-Schema ML-Schema [PEŁ+18] was developed by the W3C Machine Learning Schema Community Group. It is a top-level ontology for the description of ML algorithms, datasets, and experiments, providing terms such as task, algorithm, hyperparameter, and run. ML-Schema aims to be a commonly used standard that can be extended and specialized for more domain-specific use cases, e.g. in the area of data mining. Mappings from ML-Schema to other ontologies are also provided by the authors. The schema is meant to be used for all linked open data exports from OpenML[10] in the future. Its general applicability and existing mappings to other ontologies, as well as being a W3C recommendation make ML-Schema a good fit for adoption in HELIPORT. However, the low availability of tooling, e.g. compared to PROV, needs to be considered.

PROV-ML PROV-ML [SAL+19] combines approaches in collecting provenance information as well as ML-related metadata by extending both PROV and ML-Schema. This allows users to adequately represent domain-specific data which was created early in the experiment lifecycle, independently of ML, e.g. through data curation or conduction of lab experiments. PROV-ML distinguishes between prospective and retrospective provenance metadata, such as abstract definitions of learning workflows (prospective) and concrete executions of the workflow (retrospective). An additional feature of PROV-ML that is not part of ML-Schema is the representation of learning stages, i.e. training, validation, and test. This ontology was developed as part of the ProvLake[11] platform which the authors extended. Due to its bilateral approach towards metadata, PROV-ML suits HELIPORT better than ML-Schema alone, although lack of existing tools (apart from ProvLake) is an issue in this case as well.

ReproduceMe-ML ReproduceMe-ML [SLK21] extends REPRODUCE-ME [SK17], an ontology to describe provenance of microscopy experiments, to the area of ML, while aiming to be compatible with ML-Schema and MEX vocabulary. REPRODUCE-ME in turn uses PROV and P-Plan[12]. Due to a focus on ML experiments conducted with Jupyter Notebooks, some of the approaches taken by ReproduceMe-ML can certainly be an inspiration for future work on

[8] https://www.dublincore.org

[9] https://github.com/ewilderj/doap/wiki

[10] https://www.openml.org

[11] https://research.ibm.com/projects/provlake

[12] http://vocab.linkeddata.es/p-plan/index.html

HELIPORT. However, due to its basis in microscopy experiments, the ontologies used entail a lot of specific terms that are not applicable to most other projects which makes it unsuitable for integration in HELIPORT.

SML The Semantic Machine Learning Ontology (SML) [KMC23] is a model for describing ML models both in a human-understandable and machine-understandable way. It focuses on facilitating model selection by non-expert users by allowing them to evaluate and compare models based on their characteristics. Such characteristics can be context of the data (e.g. spatial and temporal information on collection circumstances), evaluation metrics and scores, and the application domain of the model (e.g. healthcare). While being able to compare models is certainly a concern when they are published, SML is not a good fit for documentation in HELIPORT as the context is usually already given through the project. Moreover, the experimental, iterative experimentation process is not adequately described by this ontology due to its different focus.

3.4 ML Metadata Tooling

PROV Ontology Tools Multiple of the ontologies examined in Subsection 3.3 are based on the PROV data model which is well established by now and has accumulated a large ecosystem of existing tools. This includes, among others, services such as the provenance repository ProvStore, a validator with REST API, a service to translate between different PROV representations, and a PROV Notation editor (all four on Open Provenance[13]), libraries and toolkits like the PROV Python library[14], ProvToolbox[15], and RDFLib[16] which ships with the PROV namespace builtin. Different visualization tools such as ProvViz[17] and the associated JavaScript library, as well as a variety of educational materials and guides like the PROV-PRIMER[18] also exist. This creates a great foundation for HELIPORT to build upon PROV. However, in the context of ML experiments, there is a lack of tools that provide PROV metadata to even start and make use of these tools.

ProvBook ProvBook [SK18] is an extension for Jupyter Notebook that collects provenance information of executed notebooks. Tracking is carried out on a per-cell basis and on a time scale that allows users to inspect previous contents of the cell. Rather than tracking files, artifacts, or software versions, the literal inputs and outputs are stored. All data can be exported as RDF in Turtle syntax using Jupyter Notebook-related terms of the REPRODUCE-ME ontology. Due to the provenance being recorded per cell, this tool does not match the high-level approach of HELIPORT.

MLflow2PROV MLflow2PROV [SS23] is a tool which extracts experimental metadata from MLflow and Git repositories storing the associated code. Its provenance model is compatible

[13] https://openprovenance.org/
[14] https://github.com/trungdong/prov
[15] https://github.com/lucmoreau/ProvToolbox
[16] https://github.com/RDFLib/rdflib
[17] https://provviz.com/
[18] https://www.w3.org/TR/prov-primer/

with the PROV ontology, thus existing tooling from the PROV ecosystem can be reused when utilizing the tool. MLflow2PROV focuses explicitly on extracting provenance information hidden in Git repos and using it to enrich the recorded metadata stored in MLflow. It is implemented as a command-line tool that provides subcommands to extract, merge, and transform metadata, and allows reading and writing provenance documents from/to files. The subcommands can be chained similarly to shell pipelines. Additionally, users can print statistics about their documents.

3.5 Popularity of ML Metadata Schemas

The popularity of the metadata formats examined was assessed in the context of this work by talking to ML practitioners and RSEs. RSEs were addressed in the form of a talk [PKS24] at the deRSE24 conference ("AI/ML Research Software" session) and a presentation [KMP24] as part of the HiRSE seminar series[19]. We listed the following ontologies on a slide and asked attendants about their popularity or adoption: ML-Schema, SML, DMP, OntoDM-core, Exposé, MEX, REPRODUCE-ME, ReproduceMe-ML, and PROV. As of submission of this paper, we have not received any feedback on ML ontologies after these presentations. ML practitioners were contacted through a message to the "ML@HZDR" chat room (~300 members) on the Helmholtz Cloud service[20] Mattermost. We received a single response from a person claiming to have heard of ML-Schema but to not have interacted with it themselves. Based on the reluctant feedback, we deem these ontologies and vocabularies to be virtually unknown by practitioners in the fields we reached. Nevertheless, this approach was rather qualitative and a more thorough investigation would be needed to reach a conclusion on this question.

3.6 HELIPORT

The HELIPORT software, developed in the Helmholtz Metadata Collaboration (HMC) project HELIPORT[21], is implemented as a web application that integrates with a variety of external systems that researchers use in their daily work. The source code of the application is available in [VUS$^+$23]. In addition, the "flagship" instance at HZDR is accessible to all Helmholtz employees via the Helmholtz Cloud[22].

In the following, we will explore features that are already present, and how they can be used to more comprehensibly document ML experiments, as well as how HELIPORT could be improved to adapt to this new use case. A tabular overview of the features and how they cover the requirements described in Subsection 3.1 can be found in Table 2 at the end of the section.

Data Sources "Data sources" are a resource type which can be used to refer to files or directories on network drives (SFTP, SMB) or online directories (HTTP), and are a way to register datasets in HELIPORT. In the context of ML experiments, data sources can be used to document artifacts such as training data sets, model files, or inference data. HELIPORT currently describes all data sources as generic datasets without a more specific purpose. Thus, by default,

[19] https://www.helmholtz-hirse.de/series/2024_04_11-seminar_28.html
[20] https://helmholtz.cloud
[21] https://heliport.hzdr.de
[22] https://heliport.helmholtz.cloud

training data and test data can only be described by names or tags. Similarly, models registered as datasets are not automatically enriched with metadata pertaining to the training process.

Workflows Workflows in HELIPORT refer to computational workflows (see [GCS$^+$20]) and are based on a subset of the Common Workflow Language (CWL)[23] specification. CWL allows for clear definition of the tools involved in a workflow, their interactions, as well as inputs and outputs. HELIPORT stores provenance metadata accordingly. Performance metadata is only recorded at a basic level, e.g. program runtime (wall time), so the need for extended reporting of the used compute resources can only be met in part. Using the built-in CWL execution in HELIPORT fits well established, fixed workflows where exploration is carried out by selecting from a range of methods and tweaking parameters. It is not suitable for frequently changing custom source code developed during ML experiments, or even interactive programming carried out in Juypter Notebooks. HELIPORT can also currently not be used to document, in an automated fashion, workflows run in other external workflow systems such as REANA[24], AiiDA[25], or FireWorks[26]. However, since usage of computational workflows is an important aspect to document in a FAIR experiment, alternative concepts such as those for Jupyter Notebooks presented in Subsection 3.4 need to be considered. Performance of the trained ML model is not recorded as neither CWL nor HELIPORT have knowledge about these metadata.

Documentation “Documentation” resources in HELIPORT can be used to reference any documentation relevant to the project. The registered resource is presented to the members of the project as a link without any additional functionality. Documentation resources can be used for a variety of materials, such as proposals, literature, and training materials, related to the ML experiment. Publications which are being worked on in conjunction with the project can be registered in the “ShareLaTeX” app, if a ShareLaTeX or Overleaf instance is used. Completed publications (papers, datasets, software) can be registered under “Publications”. Entries of these types also simply link to the resource.

Digital Objects Digital objects are a fundamental part of HELIPORT’s data model. They serve as a common abstraction layer and base class for all resources that can be linked in HELIPORT. Digital objects comprise basic metadata about the resource, such as owner, label, description, and category, as well as a globally unique persistent identifier (PID). Depending on the type of resource, digital objects can also contain more specific metadata, e.g. in the case of data sources, the uniform resource identifier (URI) that points to the data. The serialization process takes into account the resource type to decide which metadata schemes to use to describe the object. An overview of ontologies and vocabularies used in HELIPORT’s metadata export can be found in Table 1 and an example export is shown in Listing 1. Resource types for digital objects related to ML experiments currently do not exist. Users can enrich a digital object’s properties with semantic triples manually, either based on existing ontologies, or using their own

[23] https://www.commonwl.org
[24] https://reana.io/
[25] https://www.aiida.net/
[26] https://materialsproject.github.io/fireworks/

Table 1: Ontologies, vocabularies and schemas used for resource description in HELIPORT. Only a selection of the resource types and terms used in HELIPORT is listed.

Resource type	Schema	Terms used
Project	http://purl.org/vocab/frbr/core	`owner`
Person	http://xmlns.com/foaf/0.1/	`firstName`, `lastName`
Data source	http://purl.org/dc/dcmitype/	`Dataset`
Workflow	http://w3id.org/cwl/cwl	`CommandLineTool`, `inputs`
Documentation	http://purl.org/spar/fabio/	`PersonalCommunication`
Source code	http://purl.org/spar/fabio/	`Repository`
Any	http://purl.org/dc/terms/	`description`, `isPartOf`
	http://schema.org/	`dateDeleted`, `dateModified`
	http://rdfs.org/scot/ns	`has_tag`
	http://www.w3.org/2000/01/rdf-schema	`label`

terms. Automated metadata discovery from ML experiment tracking tools and a larger variety of workflow systems would be an improvement. Appropriate metadata formats discussed in Subsection 3.3 need to be integrated in HELIPORT accordingly. Incorporating publicly available knowledge graphs like DBpedia[27], Wikidata[28] or unHIDE[29] as metadata sources could make working with digital objects more approachable for a wider audience, if appropriate searching and linking functionality is provided.

Landing Pages Each digital object has its own landing page which can be reached by resolving the PID of the resource. Landing pages present the metadata stored in the digital object in a human-readable fashion and offer metadata exports in a variety of established formats, e.g. DataCite for bibliographic metadata, or different Resource Description Framework (RDF) serializations for semantic properties. They also make resource metadata machine-readable by implementing HTTP content negotiation, allowing clients to request any of the available metadata formats directly. Landing pages of models or datasets could, in the future, be extended with a dataset or model card view, as the concept of providing an overview of the objects' metadata is quite similar.

Digital Object Graphs Digital object graphs allow users to document and visualize relations between digital objects. This feature can be used to provide different views into a project and the resources that are part of it. Examples of facets that could be visualized are:

- Model provenance (relations between training datasets and models),

[27] https://www.dbpedia.org/
[28] https://www.wikidata.org/
[29] https://search.unhide.helmholtz-metadaten.de/

Listing 1: Metadata export initiated from a HELIPORT landing page, serialized into Turtle syntax. These metadata describe a GitHub repository that was registered as part of a HELIPORT project. The PID of the repository (https://hdl.handle.net/20.500.12865/HELIPORT.version_control.46) resolves to its landing page.

```
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix fabio: <http://purl.org/spar/fabio/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sdo: <https://schema.org/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<https://hdl.handle.net/20.500.12865/HELIPORT.version_control.46> a
    fabio:Repository ;
 rdfs:label "SOTA on Uncertainties GitHub Repository" ;
 dcterms:created "2022-06-08T14:11:48.050530+00:00"^^xsd:dateTime ;
 dcterms:isPartOf
     <https://hdl.handle.net/20.500.12865/HZDR.Projects.2022.FWCC.Project.86>
     ;
 foaf:primaryTopic "https://github.com/psteinb/sota_on_uncertainties" ;
 sdo:dateModified "2024-04-25T13:06:36.629973+00:00"^^xsd:dateTime .
```

- Computational workflows (relations between workflow runs, used software, and artifacts),
- Scientific output (relations between papers and associated data and software publications),
- Project contributors (people and institutions) and their roles.

Object graphs could also be used to show experiments and their digital twins side by side, e.g. to juxtapose experiment vs. simulation, or devices vs. virtual diagnostics. An example of this is shown in Figure 4. Another possible use-case for such graphs is the identification of upstream changes and downstream benefits of a given ML model. However, this would require relevant provenance metadata for all steps from training dataset creation to the results of an inference run of the model. Currently, digital object graphs still have to be set up manually, but once digital objects can be automatically enriched with more metadata, a default set of graphs could also be provided automatically. Moreover, digital object graphs could be used as figures in publications, addressing the need for publication assets. However, this requires the user to take a screenshot manually as they can currently not be exported as vector graphics (SVG) or graph descriptions (DOT, RDF).

Missing Features Currently, HELIPORT does not provide seamless integration from other tools. While a REST API is available that can be used to read and write most of the captured metadata, no client libraries are available to automate the process. However, a Python library is planned.[30]

[30] See presentation "Pioneering Digital Research Landscapes: Innovations at HZDR" (https://www.hzdr.de/publications/Publ-38785).

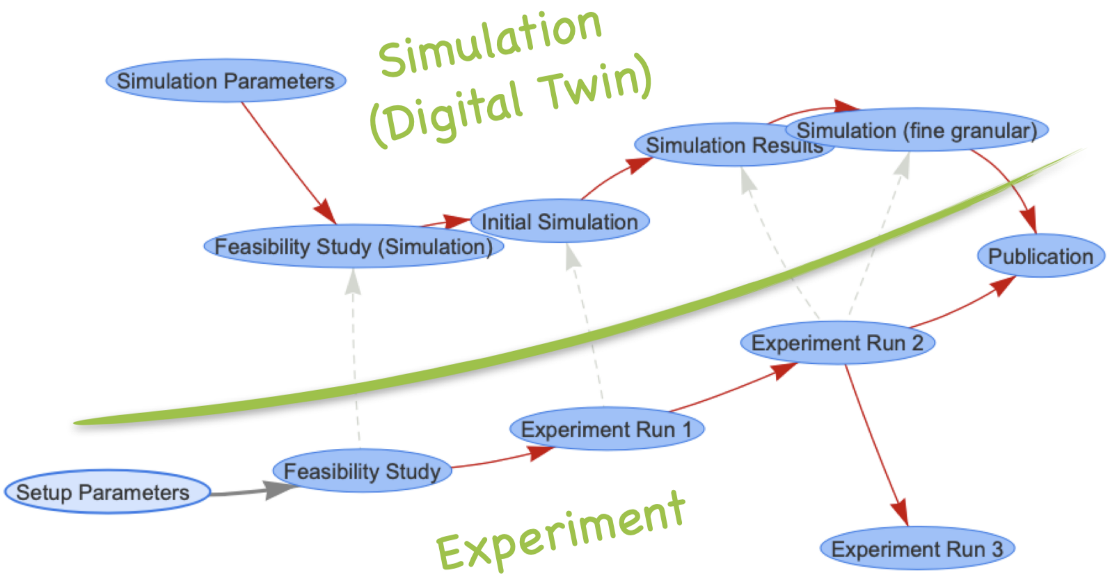

Figure 4: A digital object graph that was manually created in HELIPORT to show parallels between simulation runs and experiments. The green text and line were added in post to aid the visualization.

Table 2: An overview of the requirements identified in 3.1 and whether they are covered fully (✓), partially (❍), or not at all (✗) by HELIPORT.

Requirement	Covered	Notes
Code and configuration	❍	Only fixed workflows are covered
Environment	✗	Known partially but no traceability
Datasets and models	❍	Covered by “data sources”, but no ML context given
Compute resources used	❍	Only runtime (wall time)
Model provenance	✗	Known partially but no traceability
Cross-domain description	❍	Possible via manual use of different ontologies
Upstream changes	❍	Possible via digital object graphs
Downstream benefits	❍	Possible via digital object graphs
Collection of resources	✓	
Integration	✗	Python library planned
Automation	✗	

4 Discussion and Outlook

In this article, we have shown that existing features such as documentation resources, digital object graphs, and data sources in HELIPORT can help documenting ML experiments by embedding them in the larger context of scientific experiments, providing different views into the metadata, and building digital twins. But to make this potential usable for scientists, metadata acquisition, especially of computational and machine learning workflows, needs to be improved. Development of a HELIPORT Python library will be a serviceable approach in this direction. Provision of more, potentially arbitrary metadata might lead to new requirements in HELIPORT's data model. The areas of semantic web and knowledge graphs offer a variety of paths that can be explored in this regard.

Obtaining high-quality, rich experiment metadata from tools that are well established and used by ML practitioners is a large concern. Changing over to different tools that improve metadata extraction is not a consideration as we do not want to disrupt researchers' personal workflows. While some metadata can be extracted from existing workflows, e.g. via instrumentation of the code, subsequent translation of the metadata into a fitting metadata scheme will be required. This will establish a common interface layer between HELIPORT and ML experiment tracking tools. A good candidate for this layer would be PROV-ML.

Looking forward, we will approach further development of HELIPORT with the mentioned issues in mind, and select concrete use cases from the ML domain to design and implement new features.

Acknowledgements: The project HELIPORT (ZT-I-PF-3-021) was funded by the "Initiative and Networking Fund of the Helmholtz Association" in the framework of the first “Helmholtz Metadata Collaboration (HMC)” project call 2021.

References

[Bie20] L. Biewald. Experiment Tracking with Weights and Biases. 2020. Software available from wandb.com.
https://www.wandb.com/

[CAB+20] M.-E. Couprie, T. André, F. Blache, F. Bouvet, Y. Dietrich, J.-P. Duval, M. El-Ajjouri, A. Ghaith, C. Herbeaux, N. Hubert, C. Kitégi, M. Khojoyan, M. Labat, N. Leclercq, A. Lestrade, A. Loulergue, O. Marcouillé, F. Marteau, D. Oumbarek-Espinos, P. Rommeluére, M. Sebdaoui, K. Tavakoli, M. Valléau, S. Corde, J. Gautier, J. P. Goddet, O. Kononenko, G. Lambert, A. Tafzi, K. T. Phuoc, C. Thaury, S. Bielawski, E. Roussel, C. Szwaj, I. Andriyash, V. Malka, S. Smartsev. Progress towards laser plasma based free electron laser on COXINEL. *Journal of Physics: Conference Series* 1596(1):012040, July 2020.
doi:10.1088/1742-6596/1596/1/012040

[EMN+15] D. Esteves, D. Moussallem, C. B. Neto, T. Soru, R. Usbeck, M. Ackermann, J. Lehmann. MEX vocabulary: a lightweight interchange format for machine learn-

ing experiments. In *Proceedings of the 11th International Conference on Semantic Systems*. Pp. 169–176. ACM, Vienna Austria, Sept. 2015.
doi:10.1145/2814864.2814883

[GCS+20] C. Goble, S. Cohen-Boulakia, S. Soiland-Reyes, D. Garijo, Y. Gil, M. R. Crusoe, K. Peters, D. Schober. FAIR Computational Workflows. *Data Intelligence* 2(1-2):108–121, Jan. 2020.
doi:10.1162/dint_a_00033

[HSA+22a] H. Hessenkemper, S. Starke, Y. Atassi, T. Ziegenhein, D. Lucas. Bubble identification from images with machine learning methods. *International Journal of Multiphase Flow* 155:104169, Oct. 2022.
doi:10.1016/j.ijmultiphaseflow.2022.104169

[HSA+22b] H. Hessenkemper, S. Starke, Y. Atassi, T. Ziegenhein, D. Lucas. Software for Bubble identification from images with machine learning methods. Aug. 2022.
doi:10.14278/rodare.1470

[KŁd+15] C. M. Keet, A. Ławrynowicz, C. d'Amato, A. Kalousis, P. Nguyen, R. Palma, R. Stevens, M. Hilario. The Data Mining OPtimization Ontology. *Journal of Web Semantics* 32:43–53, May 2015.
doi:10.1016/j.websem.2015.01.001

[KMC23] L. Kallab, E. Mansour, R. Chbeir. SML: Semantic Machine Learning Model Ontology. In Zhang et al. (eds.), *Web Information Systems Engineering – WISE 2023*. Volume 14306, pp. 896–911. Springer Nature Singapore, Singapore, 2023.
doi:10.1007/978-981-99-7254-8_70

[KMP24] O. Knodel, S. Müller, D. Pape. Facilitating Research Data Management with HELIPORT. Apr. 2024.
doi:10.5281/zenodo.10993243

[KVU+20] O. Knodel, M. Voigt, R. Ufer, D. Pape, M. Lokamani, S. E. Müller, T. Gruber, G. Juckeland. HELIPORT: A Portable Platform for {FAIR Workflow | Metadata | Scientific Project Lifecycle} Management and Everything. In *Proceedings of the 4th International Workshop on Practical Reproducible Evaluation of Computer Systems*. Pp. 9–14. ACM, Virtual Event Sweden, June 2020.
doi:10.1145/3456287.3465477

[MWZ+19] M. Mitchell, S. Wu, A. Zaldivar, P. Barnes, L. Vasserman, B. Hutchinson, E. Spitzer, I. D. Raji, T. Gebru. Model Cards for Model Reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. Pp. 220–229. ACM, Atlanta GA USA, Jan. 2019.
doi:10.1145/3287560.3287596

[PEŁ+18] G. C. Publio, D. Esteves, A. Ławrynowicz, P. Panov, L. Soldatova, T. Soru, J. Vanschoren, H. Zafar. ML-Schema: Exposing the Semantics of Machine Learning with

Schemas and Ontologies. July 2018.
doi:10.48550/arXiv.1807.05351

[PKS24] D. Pape, O. Knodel, S. Starke. Documenting ML Experiments in HELIPORT. Mar. 2024.
doi:10.5281/zenodo.10807608

[PSD14] P. Panov, L. Soldatova, S. Džeroski. Ontology of core data mining entities. *Data Mining and Knowledge Discovery* 28(5-6):1222–1265, Sept. 2014.
doi:10.1007/s10618-014-0363-0

[SAL+19] R. Souza, L. Azevedo, V. Lourenço, E. Soares, R. Thiago, R. Brandão, D. Civitarese, E. V. Brazil, M. Moreno, P. Valduriez, M. Mattoso, R. Cerqueira, M. A. S. Netto. Provenance Data in the Machine Learning Lifecycle in Computational Science and Engineering. Oct. 2019.
doi:10.48550/arXiv.1910.04223

[SEZ+22] S. Starke, J. Eulitz, A. Zwanenburg, E. G. Troost, M. Krause, A. Lühr, S. Löck. Convolutional neural networks predict the linear energy transfer for proton-beam radiotherapy of patients with brain tumours. In *Medical Imaging with Deep Learning*. 2022.
https://openreview.net/forum?id=ue4_3NG344g

[SK17] S. Samuel, B. König-Ries. REPRODUCE-ME: Ontology-Based Data Access for Reproducibility of Microscopy Experiments. In Blomqvist et al. (eds.), *The Semantic Web: ESWC 2017 Satellite Events*. Volume 10577, pp. 17–20. Springer International Publishing, Cham, 2017.
doi:10.1007/978-3-319-70407-4_4

[SK18] S. Samuel, B. König-Ries. ProvBook: Provenance-based Semantic Enrichment of Interactive Notebooks for Reproducibility. *Proceedings of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks*, 2018.
https://ceur-ws.org/Vol-2180/paper-57.pdf

[SLK21] S. Samuel, F. Löffler, B. König-Ries. Machine Learning Pipelines: Provenance, Reproducibility and FAIR Data Principles. In Glavic et al. (eds.), *Provenance and Annotation of Data and Processes*. Volume 12839, pp. 226–230. Springer International Publishing, Cham, 2021.
doi:10.1007/978-3-030-80960-7_17

[SS23] M. Schlegel, K.-U. Sattler. Extracting Provenance of Machine Learning Experiment Pipeline Artifacts. In Abelló et al. (eds.), *Advances in Databases and Information Systems*. Volume 13985, pp. 238–251. Springer Nature Switzerland, Cham, 2023. Series Title: Lecture Notes in Computer Science.
doi:10.1007/978-3-031-42914-9_17

[VS10] J. Vanschoren, L. Soldatova. Exposé: An ontology for data mining experiments. In *International workshop on third generation data mining: Towards service-oriented knowledge discovery (SoKD-2010)*. Pp. 31–46. 2010.
https://lirias.kuleuven.be/retrieve/118733

[VUS+23] M. Voigt, R. Ufer, W. Schacht, O. Knodel, D. Pape, M. Lokamani, S. Müller, T. Gruber, J. Kelling. HELIPORT (HELmholtz ScIentific Project WORkflow PlaTform). June 2023.
doi:10.14278/rodare.2334

[ZCD+18] M. Zaharia, A. Chen, A. Davidson, A. Ghodsi, S. A. Hong, A. Konwinski, S. Murching, T. Nykodym, P. Ogilvie, M. Parkhe et al. Accelerating the machine learning lifecycle with MLflow. *IEEE Data Eng. Bull.* 41(4):39–45, 2018.
http://sites.computer.org/debull/A18dec/p39.pdf