Advanced Engineering Materials

ADVANCED ENGINEERING MATERIALS
www.aem-journal.com

REVIEW

Supporting AI Readiness Through Digital Workflows in Materials Science

Marian Bruns[1] | Lateef Abdul[2] | Stephan Barth[3] | Martin Glauer[4] | Manuel Günther[3] | Fabian Neuhaus[4] | Jan-Ole Perschewski[4] | Thomas Bjarsch[5] | Henrik Abel[6] | Johannes Bosch[7] | Niklas Meyer[8] | Peter Hofmann[9] | Marc Drechsler[10] | Alexander Dyck[11] | Dhanunjaya Kumar Nerella[12] | Marvin Tegeler[12,13] | Oleg Shchyglo[12,13] | Ingo Steinbach[12,13] | Timo Klecker[14] | Henk Birkholz[15] | Ehsan Borzabadi Farahani[16] | Bernard Fedelich[16] | Cetin Haftaoglu[16] | Tarakeshwar Lakshmipathy[16] | Elena Garcia Trelles[17] | Christoph Schweizer[17] | Reza Darvishi Kamachali[16,18] | Lukas Koschmieder[19] | Janin Eiken[19] | Markus Apel[19] | Sung-Min Wi[20] | Jiangdong Zhao[20] | Andreas Fezer[20] | Nicolas Christ[17] | Patrick Schneider[21] | Eldor Urinov[7] | Veit Königer[7] | Volker Knoblauch[7] | Janik Harter[22] | Thomas Seifert[22] | Thitichai Janpheng[22] | Florian Fuchs[23,24] | Fabian Teichert[23,24] | Jörg Schuster[23,24] | Anke Bardehle[25] | Andreas Limper[25] | Ulrich Giese[26] | Carlos Lucke[26] | Mario Beiner[27] | Christoph Gögelein[28] | Lisa Leuchtenberger-Engel[25] | Christian Hopmann[25] | Felix Wentzien[29] | Heike Wittek[26] | Benjamin Klie[26] | Alexander Röhrs[30] | Ludger Overmeyer[29] | Achraf Atila[16] | Marcel Sadowski[31] | Leopold Talirz[31] | Jan Janssen[1] | Jesper R. Pedersen[32] | Peter Beck[32] | Tejs Vegge[32,33] | Ivano E. Castelli[32] | Wolfgang Wenzel[2] | Tilmann Hickel[16] | Jörg Schaarschmidt[2]

[1]Computational Materials Design (CM), Max-Planck-Institute for Sustainable Materials, Düsseldorf, Germany | [2]Karlsruhe Institute of Technology (KIT), Institute of Nanotechnology (INT), Eggenstein-Leopoldshafen, Germany | [3]Fraunhofer Institute for Electron Beam and Plasma Technology (FEP), Dresden, Germany | [4]Otto von Guericke University Magdeburg, Magdeburg, Germany | [5]Fraunhofer Institute for Casting, Composite and Processing Technology (IGCV), Augsburg, Germany | [6]Enari GmbH, Garching, Germany | [7]Aalen University of Applied Sciences, Aalen, Germany | [8]Novicos GmbH, Hamburg, Germany | [9]Fraunhofer Institute for Digital Media Technology (IDMT), Ilmenau, Germany | [10]Cotesa GmbH, Mittweida, Germany | [11]Robert Bosch GmbH, Stuttgart, Germany | [12]Ruhr-Universität Bochum, ICAMS, Bochum, Germany | [13]OpenPhase Solutions GmbH, Bochum, Germany | [14]DECOIT GmbH & Co., Bremen, Germany | [15]Leibniz-Institute for Materials Engineering – IWT, Bremen, Germany | [16]Bundesanstalt für Materialforschung und -prüfung (BAM), Berlin, Germany | [17]Fraunhofer Institute for Mechanics of Materials (IWM), Freiburg, Germany | [18]Institute of Materials Physics, University of Münster, Münster, Germany | [19]Access e.V., Aachen, Germany | [20]Material Testing Institute University of Stuttgart, Stuttgart, Germany | [21]Siemens AG, München, Germany | [22]Institute for Digital Engineering and Production (IDEeP), Offenburg University of Applied Sciences, Offenburg, Germany | [23]Fraunhofer Institute for Electronic Nano Systems (ENAS), Chemnitz, Germany | [24]Center for Materials, Architectures and Integration of Nanomembranes (MAIN), Chemnitz University of Technology, Chemnitz, Germany | [25]Institute for Plastics Processing (IKV), RWTH Aachen University, Aachen, Germany | [26]DIK Deutsches Institut für Kautschuktechnologie e.V., Hannover, Germany | [27]Fraunhofer Institute for Microstructure of Materials and Systems (IMWS), Halle, Germany | [28]ARLANXEO Deutschland GmbH, Dormagen, Germany | [29]Institute of Transport and Automation Technology (ITA), Leibniz Universität Hannover, Garbsen, Germany | [30]deepIng business solutions GmbH, Hannover, Germany | [31]Computational Materials Engineering, SCHOTT AG, Mainz, Germany | [32]Department of Energy Conversion and Storage, Technical University of Denmark, Kgs. Lyngby, Denmark | [33]CAPeX Pioneer Center for Accelerating P2X Materials Discovery, Kgs. Lyngby, Denmark

Correspondence: Tilmann Hickel (tilmann.hickel@bam.de) | Jörg Schaarschmidt (joerg.schaarschmidt@kit.edu)

Received: 3 June 2026 | **Revised:** 20 August 2026 | **Accepted:** 24 August 2026

Keywords: AI-readiness | multiscale simulations | workflows

ABSTRACT

Materials science produces heterogeneous data across experiments, simulations, and industrial processes that often remain bound to local formats, manual procedures, and project-specific software. Digital workflows address this fragmentation through explicit, repeatable, and machine-actionable pipelines. This article examines 13 workflow contributions from the second and third funding phases of the MaterialDigital initiative. The contributions cover data acquisition and FAIR storage, simulation automation,

multiscale integration, and AI/ML-driven optimization. Here, AI-readiness denotes the documented capacity of workflows and their artifacts to be reliably interpreted, executed, assessed, reused, and, where intended, invoked or adapted by automated systems; it does not require the direct application of artificial intelligence. Eight comparison aspects capture data and metadata, reusable artifacts, orchestration, robustness, cross-scale coupling, transfer validation, learning and optimization, and adaptive or agent-accessible operation. By distinguishing demonstrated capabilities from plans, the comparison shows how workflows support AI-ready research both through integrated AI methods and through structured, traceable, and reusable pipelines. Across the projects, stable data structures, persistent artifacts, executable orchestration, and scientific validation recur as foundations, whereas adaptive operation and agent-accessible interfaces remain specialized. Wider interoperability would additionally benefit from shared interfaces and explicit, portable descriptions of workflow artifacts.

1 | Introduction

The digital transformation of materials science is driven by the growing amount and complexity of data produced in experiments, simulations, and industrial processes. This development creates new opportunities for data-driven materials design, but it also exposes a recurring problem: data often remain bound to local file formats, project-specific scripts, manual processing steps, or isolated databases. Under these conditions, results are difficult to compare and reproduce, and validation becomes cumbersome. Limited interoperability between data and software tools further hinders reuse across institutional and disciplinary boundaries.

The MaterialDigital (MD) initiative addresses these issues by developing frameworks and software solutions for the digitalization of materials science in Germany. It does so through a variety of individual projects covering different materials, manufacturing processes, experimental and characterization methods, and simulation approaches. The Platform MD (PMD) is the central coordinating entity within the initiative. It develops and provides prototypical components and services for a shared digital ecosystem, including ontologies such as the PMD Core Ontology (PMDco) [1], the workflow environments SimStack [2] and pyiron [3], further software components, and publication platforms such as the PMD Workflow Store and the MaterialDigital DataPortal.

The present article extends the earlier cross-project workflow review [4] by focusing on projects from the second and third MD funding phases. Ten contributions originate from the second funding phase, whose projects started between September 2022 and January 2024 and are approaching the end of their funding periods [5]. DiMad and DiMoGraph were already represented in the earlier article and provide updated accounts of their workflow implementations. The three third-phase projects presented here started in 2025 and remain at an early stage. Their sections therefore necessarily place greater emphasis on concepts, architectures, and planned capabilities. Funding phase and contribution status are identified in Table 1 and are taken into account when distinguishing demonstrated functionality from outlooks.

The structure of a large, multiproject initiative such as MD benefits from a diversity of requirements and solutions. The participating projects differ in the state and organization of their data, software environments, experimental routines, simulation methods, and requirements for automation or confidentiality. Some workflows rely on commercial engineering tools, while others are built on open-source environments. Some projects require high-performance computing (HPC) systems or graph databases, while others depend mainly on laboratory-specific data-acquisition pipelines. As a result, no single technical stack can cover this diversity.

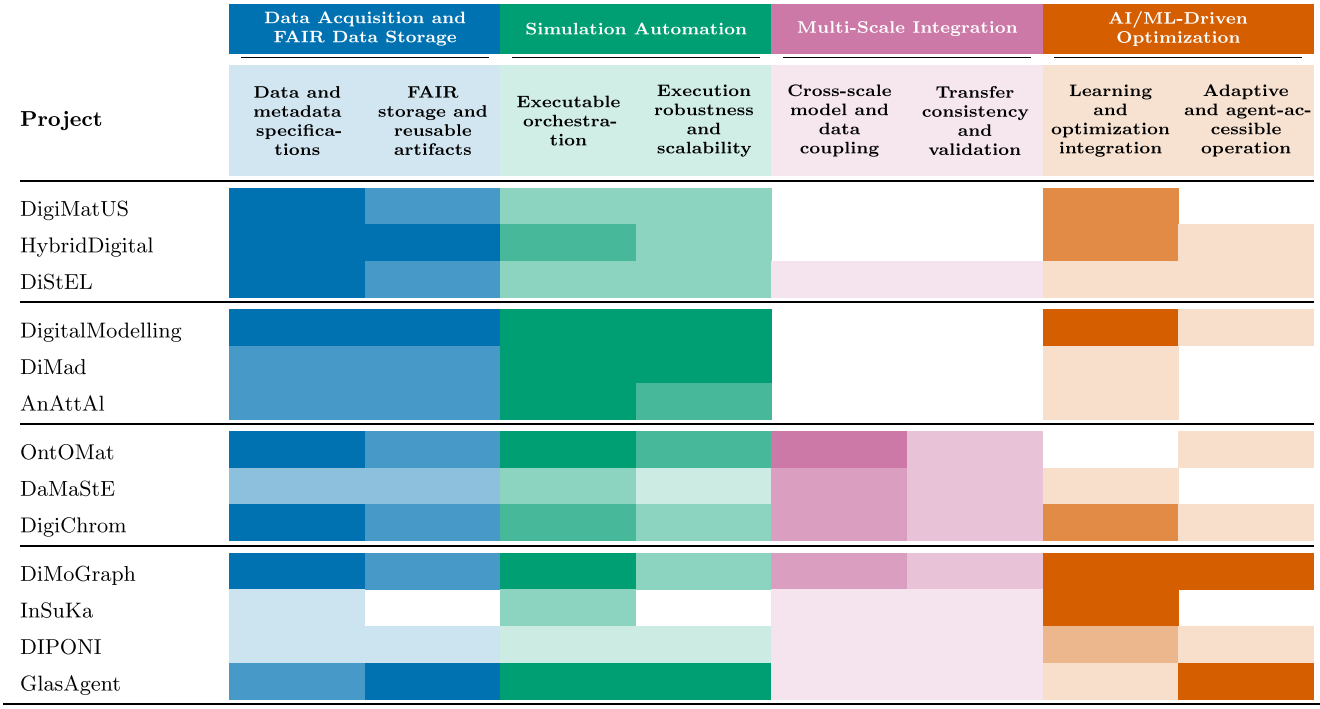
1.1 | Workflows as Project-Level Abstractions

Workflows provide project-level abstractions over these heterogeneous tools and processing steps. They describe how data are acquired, transformed, enriched, simulated, analyzed, stored, and reused. In this article, the term refers to structured and repeatable sequences of tasks that are required to achieve a defined result. These tasks may include experimental data acquisition, raw-data preprocessing, semantic annotation, computational simulation chains, parameter identification, model calibration, machine-learning pipelines, or the publication of reusable workflow components. Where workflows additionally adopt compatible interfaces, data structures, and artifact descriptions, these abstractions can support reuse beyond the original project and contribute to initiative-wide interoperability. The contributions presented here should nevertheless not be interpreted as implementations of one uniform abstraction layer; many remain tailored to their respective scientific and technical contexts.

Digital workflows reduce manual and frequently undocumented processing by making scientific procedures machine-actionable, reproducible, and reusable. They can additionally integrate heterogeneous software, automate parameter studies, provide access to HPC resources, and capture intermediate and final results. These general benefits and their implementation within MD were discussed in detail in previous works [4, 6].

Within MD, pyiron and SimStack are the canonically supported workflow environments. They provide complementary approaches to workflow development and execution and are used by several of the projects. Although both originated in efforts to streamline atomistic and multiscale simulations, the contributions in this article demonstrate their wider applicability. Although the article focuses on MD, the workflow concepts and implementations are intended for uptake by the wider materials-science community in Germany and internationally. This broader scope is reflected in the recently launched MaterialsCommons initiative [7], which builds on MD and other European activities. Workflows form a central domain of MaterialsCommons, with an emphasis on interoperability between established environments rather than on a single workflow manager. The conclusions illustrate this extension

TABLE 1 | Documented AI-readiness profiles of 13 application-oriented project contributions.



Note: The project groups reflect their principal workflow focus, and the row order matches the article sequence, while every contribution is shown across all eight aspects. Each pair of aspect columns uses the base color of the corresponding thematic category. Increasing color intensity denotes increasing documented implementation and integration: no fill = not documented in the contribution; light = mentioned, planned, indirect, or isolated; low–medium = partially implemented; high-medium = substantially implemented and important; strong = defining and deeply integrated. The profiles are intended as a visual guide to the contributions rather than as grades, rankings, software certifications, maturity measurements, or assessments of scientific quality. The scoring procedure and explicit numeric levels are provided in Table S1; the comparison therefore does not depend on distinguishing color shades.

through PerQueue as one example of a workflow framework contributing to MaterialsCommons alongside AiiDA and other community solutions.

1.2 | Supporting AI-Readiness Through Workflows in MD

The MD projects presented illustrate four principal roles of workflows. *Data Acquisition and FAIR Data Storage* covers the transformation of heterogeneous data into machine-readable resources and are represented by *data and metadata specifications* and *FAIR storage and reusable artifacts*. *Simulation Automation* covers machine-executable scientific procedures and is represented by *executable orchestration* and *execution robustness and scalability*. *Multiscale integration* covers the transfer of information between models or scales and is represented by *cross-scale model and data coupling* and *transfer consistency and validation*. *AI/ML-driven optimization* covers the direct integration of learning, prediction, optimization, or automated decision-making and is represented by *learning and optimization integration* and *adaptive and agent-accessible operation*. These categories describe the principal focus of each contribution and are not mutually exclusive. Table 1 summarizes the documented AI-readiness profiles of the thirteen application-oriented contributions across the eight corresponding aspects.

In this article, AI-readiness denotes the degree to which a workflow and the artifacts generated by it can be reliably interpreted, executed, assessed, and reused by automated computational systems. Such systems may include conventional machine-learning pipelines, surrogate models, active-learning and optimization algorithms, process-control systems, neuro-symbolic methods, or large-language-model-based agents. A workflow does not need to execute an AI method itself to contribute to AI readiness. For example, a workflow may generate structured and traceable training, validation, or simulation data for subsequent use by an AI system. Conversely, the inclusion of a machine-learning model does not by itself make the surrounding data and execution pipeline AI-ready.

Table 1 is intended as a visual guide to the different emphases and documented implementation states across the contributions. The scoring procedure and the numerical levels underlying the color intensities are provided in the Supporting Information.

The profiles distinguish documented implementations from plans and outlooks; absence of a fill indicates only that the corresponding capability is not documented in the contribution.

The present reflection focuses primarily on the technical readiness of the workflow pipeline rather than on the complete semantic conditioning of every scientific dataset. Ontologies and semantic annotations contribute to machine interpretability through explicit metadata, identifiers, and provenance. The detailed design, alignment, and validation of domain ontologies, however, are not evaluated comprehensively here. Technical, semantic, and conceptual interoperability are discussed in greater depth, with particular emphasis on integrating workflow

systems and knowledge graphs (KG) in a separate publication by Janssen et al. [8].

2 | Workflow Approaches Across the MD Initiative

Workflows are a major component of the MD projects presented here, but each project addresses a distinct materials-science application with specific challenges and solutions. Table 2 summarizes their application domains, principal technical components, and primary workflow contributions.

2.1 | DigiMatUS: Neuro-Symbolic Machine Learning for Thin-Film Coating Processes

MD Funding Phase 2

2.1.1 | Introduction

Thin-film materials are used in many fields, including microelectronics, sensors, optics, biology, and medical technology. The coating processes that are used for these films can be very expensive in terms of time and resources, and specific applications may require very particular film properties. Furthermore, determining suitable source-material parameters, machine settings, and process conditions for producing a film with the desired properties is nontrivial because many of these factors are intertwined.

The *DigiMatUS* project therefore aims to reduce the time, cost, and material effort required to develop thin-film materials with defined target properties. To this end, the project is developing a machine-learning-powered system that determines the process parameters required to produce a film with a specific set of properties.

Piezoelectric thin films for high-resolution ultrasound microscopy serve as the main application example. The small microelectronic structures that are investigated in this use case require high operating frequencies of the transducers that can only be achieved by particularly thin films with a well-oriented and homogeneous crystalline structure.

2.1.2 | Approach and Methods

This application presents several challenges. The expensive nature of production runs greatly limits the availability of data

TABLE 2 | Application domains, principal technical components, and primary workflow contributions of the 13 MaterialDigital projects discussed in this article.

Project (phase)	Materials or process domain	Workflow and software components	Primary workflow contribution
1. DigiMatUS (2)	AlN; AlScN; thin films	Python; ontology-based database; HDF5	Data acquisition; neuro-symbolic ML
2. HybridDigital (2)	CFRP-metal hybrids	Python; Docker; triplestore; Angular	FAIR data pipeline; digital part record; ML prediction
3. DiStEL (3)	Steel components; automotive value chain	pyiron; OntoDocker; event-driven pipeline	Semantic data pipeline; multiscale simulation
4. DigitalModelling (2)	IN718; constitutive models; material cards	pyiron; Simul; SciPy; knowledge graph	Parameter identification; validation; material cards
5. DiMad (2*)	Additive manufacturing; steel alloys	pyiron; SimStack; Jupyter; Jinja; MicPy; MICRESS	Parameter studies; legacy-code integration
6. AnAttAl (2)	Al alloys; resistance spot welding	pyiron; Abaqus; Fortran solver	GUI support; reproducibility; solver coupling
7. OntOMat (2)	Fiber-reinforced composites	pyiron; Abaqus; KG API; SLURM	Multiscale FE homogenization; KG coupling
8. DaMaStE (2)	Li-ion batteries; NCM622 cathodes	GeoDict; battery database	3D microstructures; transport simulation
9. DigiChrom (2)	Cr(III)-based coatings	pyiron; Python; Abaqus; graph database	Microstructure generation; parameter optimization
10. DiMoGraph (2*)	Graphene macromaterials	pyiron; Jupyter Notebook; owlready2	Active learning; surrogate modeling
11. InSuKa (2)	Rubber compounds; mixing processes	InSuKa app; laboratory mixing data; mathematical process models	Recipe and process optimization; prediction of mixing instructions
12. DIPONI (3)	Rubber extrusion; polymer processing	pyiron (planned)	AI control; quality prediction
13. GlasAgent (3)	Specialty glasses	pyiron; amorphousPy; LAMMPS; LLM agent	Atomistic workflows; agent interface

Note: Project numbering follows the article sequence. Numbers in parentheses denote the MaterialDigital funding phase; an asterisk identifies a substantially updated account of aproject already represented in the preceding workflow article [4]. Phase 3 projects started in 2025 and are therefore at an early stage. The overview is descriptive; documented implementation status and AI-readiness profiles are summarized separately in Table 1.

points that can be used to train a pure machine learning (ML) model. Further, it may be the case that the set of features tracked by production machines may not suffice to predict the correct process parameters with the desired precision. Yet, despite their complexity, the coating processes used to produce these thin films are physical processes. A machine-learning model designed to predict the process parameters for a desired film also has to learn the physical interdependencies between different parameters. These interdependencies follow natural laws that are, to some extent, known and describable by experts. This expert knowledge can simplify the learning task when integrated into the machine-learning model architecture, thereby reducing the amount of data needed for training.

To achieve this goal, the DigiMatUS team designed a formal representation for coating processes. The Coating Ontology (CoatO) [9] is an ontology based on the basic formal ontology (BFO). It covers core concepts that are required to describe coating processes, such as machines and their parts, as well as materials and their properties. This ontology is then used as the foundation for a flexible data model that allows the integration of coating-process data from different sources. A second contribution is a formal description of physical properties of coating processes and the ways in which they interconnect. The connections in this dependency model can range from concrete formulas to more general "rules of thumb." The data stored in the ontology-based data model are then used to train a neuro-symbolic system that incorporates knowledge from the dependency model. Given the limited availability of data in this domain, the project extends a rule-based system [10, 11] with complex terms. These terms incorporate expert knowledge and established scientific laws directly into the system, thereby reducing the amount of training data needed.

2.1.3 | *Relevant Workflows*

Different steps of this project involve relevant workflows. An overview of the workflow pipelines is shown in Figure 1, from the experimental workflows at the top to the data acquisition and processing workflows, as well as the ontological workflows at the bottom of the figure, culminating in the ML aspect for prediction of process and film properties.

The ontological work started in this project is currently focused on the scope of this project: the deposition of highly oriented crystalline thin films via physical vapor deposition (PVD) processes, specifically reactive magnetron sputter processes. The CoatO is, however, designed to be extensible to other kinds of coating processes and attachable to the PMDco V3. Once an ontology is used across different teams and applications, a collaborative ontology workflow is required. We therefore also founded and contributed to the definition of a collaborative ontology development workflow [12].

Data acquisition from the coating machines, process logs, and film analytics is performed using purpose-built Python scripts. The raw data sources and components produced by experimental work are semantically complex, diverse in structure, and heterogeneous in storage format. Thus, the data acquisition workflow involves several steps. Raw data sources were classified and associated with the related process and substrate units. Then, the data are read from the respective sources, transformed, and processed. This includes merging of data from distributed sources, alignment of representation and data types, and, where applicable, derivation of meaningful scientific metrics via aggregation to reduce complexity. Finally, the results are appropriately decomposed and inserted into the database. The data schema of this

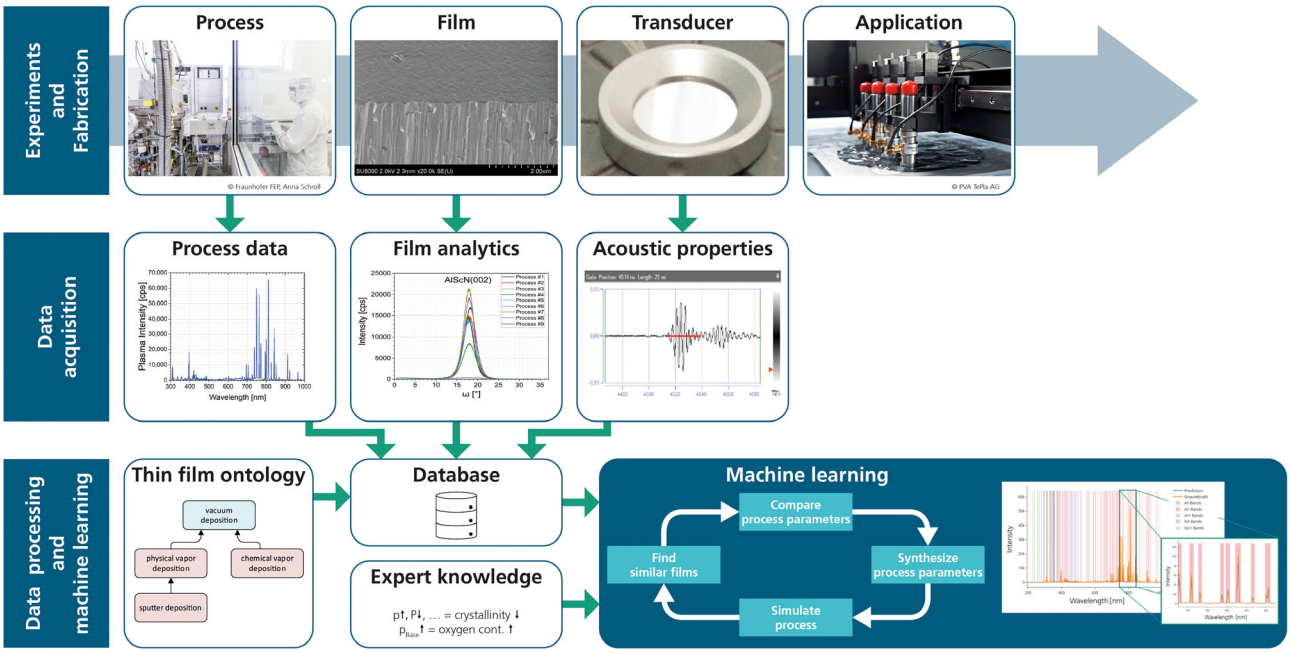


FIGURE 1 | DigiMatUS project overview with experimental and use-case visualization (top), data generation from experimental analysis (middle), and data processing as basis for the training of a neuro-symbolic machine learning model (bottom).

database is aligned with the CoatO and, therefore, allows flexible data storage. Finally, the data are loaded into a structured intermediate storage format, such as HDF5, before being passed to our ML models.

2.2 | HybridDigital: FAIR Data Pipeline for Carbon Fiber Reinforced Plastics (CFRP)-Metal Hybrid Materials

MD Funding Phase 2

2.2.1 | Motivation and Use Cases

Hybrid materials combine two or more material classes to leverage complementary properties. For CFRP–metal hybrids, this typically means combining the low density and high strength of CFRP with the low cost and high stiffness of metal [13]. Typical application domains include the automotive and aerospace industries. Processing hybrid materials comprises manufacturing and preparing the individual constituents and joining them, for example, via mechanical fasteners (bolts, screws, pins) or adhesives [14].

Design is complex. It requires handling many interacting parameters, such as CFRP ply layup, joining-element characteristics (e.g., adhesive type, bolt diameter), and geometric features (e.g., overlap length, hole diameter). Experimental campaigns are therefore time-consuming and expensive [15]. Building transferable knowledge is difficult because the process chain spans several domains (materials science, mechanical engineering, and chemistry) [16]. This leads to heterogeneous data formats and inconsistent descriptions, which hinders holistic analyses across design and processing.

Figure 2 summarizes five industry-relevant use cases: (1) trace the manufacturing history of parts, (2) identify commonalities and differences between manufactured batches, (3) identify correlating factors across the process chain (e.g., process settings vs. material properties), (4) predict geometric properties for the design phase, and (5) predict process settings to achieve target material properties.

To address these use cases, we manufacture specimens with systematically varied geometries and process settings (e.g., bolt diameter, overlap length, curing temperature, joining method, and surface treatment). We characterize tensile properties (e.g., tensile strength and stiffness) and collect data along the full process chain (e.g., raw material properties and process parameters). We then run structural finite-element simulations, validate them against experiments, and generate additional parameter combinations not covered experimentally (e.g., further overlap lengths or adherend stiffness values). All experimental and simulation raw files are stored in shared storage and processed by a pipeline that transforms them into FAIR data. An ontology defines the schema; raw data are mapped by creating and linking instances. The resulting FAIR dataset supports statistical analysis, correlation analysis, and ML models for prediction.

2.2.2 | Implemented Workflow

Figure 3 shows the workflow from raw files to FAIR data. The Docker application implementing the workflow is available in the HybridDigital GitLab repository [17]. The pipeline separates input extraction from semantic modeling.

2.2.2.1 | Parsers. Parser scripts (GitLab-versioned Python scripts) ingest specific raw inputs (e.g., spreadsheets with autoclave temperatures), extract relevant values, and export them as JSON. This unifies heterogeneous file types (CSV, XLSX, and TXT) into a common intermediate format. Versioning enables controlled parser updates (e.g., extracting additional variables) and supports provenance tracking in the resulting FAIR data.

2.2.2.2 | Connectors. Connector scripts (GitLab-versioned Python scripts) consume JSON outputs and check for required information (e.g., specimen ID and time series of force and displacement). If complete, a connector creates instances of a class and links them according to the modeling approach. For example, it creates a `TensileTest` instance, resolves the corresponding specimen instance via specimen ID, and links both. Versioning supports changes in the modeling approach while preserving traceability. The parser/connector split reduces implementation effort because a connector can be reused across sources (e.g., experiments vs. simulations) as long as both parsers output the same JSON structure. Each connector writes its triples into a separate named graph.

2.2.2.3 | Ontologies and Provenance. We use the BFO as the top-level ontology and the Common Core Ontologies (CCO) as the mid-level ontology. The pipeline records the full provenance

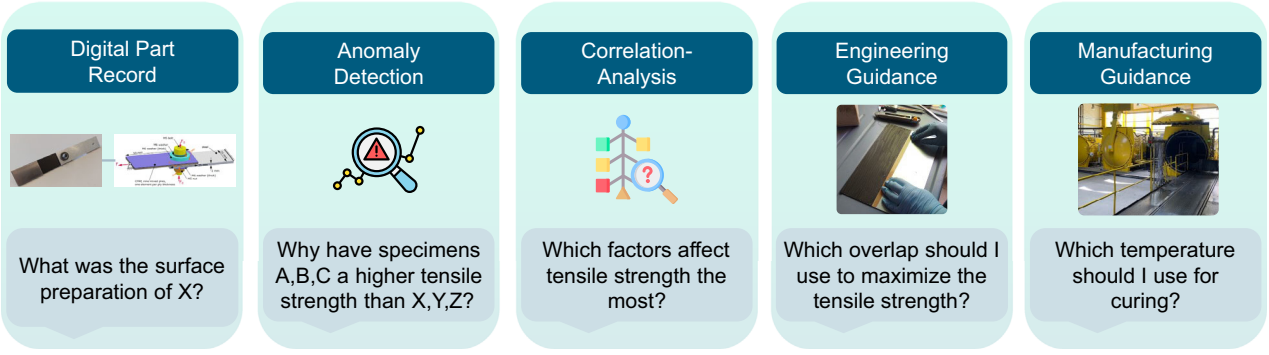


FIGURE 2 | Addressed use cases with exemplary questions to be answered.

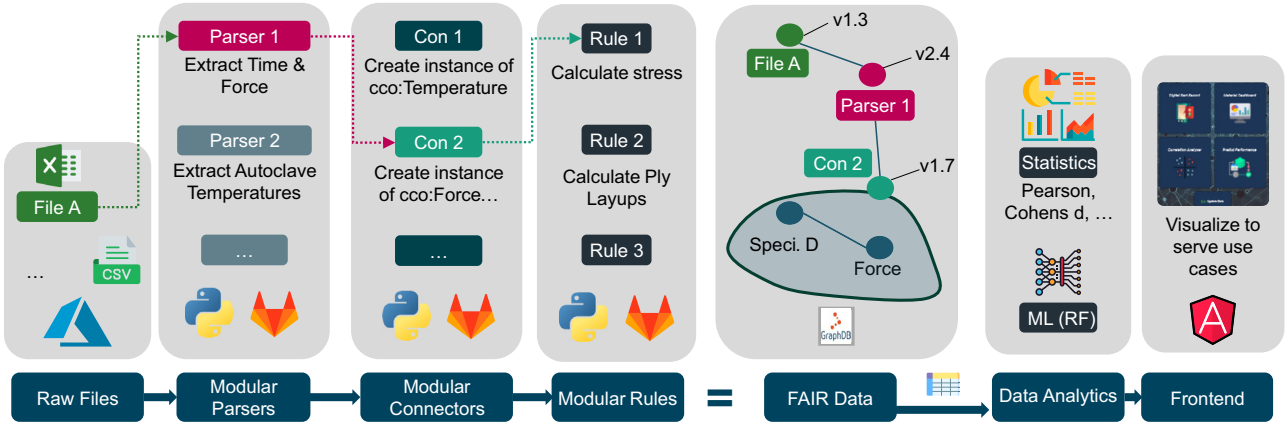


FIGURE 3 | Pipeline creating FAIR data in a triplestore, serving as the backbone for statistical analysis and ML to address the use cases.

chain from raw files to instantiated knowledge-graph content. Each raw file is represented by an instance. If a raw file changes, the pipeline creates a new raw-file instance and links it to the previous version. The same mechanism applies to parser and connector scripts: each executed script has an instance, and updated script versions are linked to prior versions.

2.2.2.4 | *Querying the Latest State.* For most queries (e.g., retrieving material properties for a specimen), only the most recent version of each provenance chain is relevant: the latest raw file, the latest parser version that parsed it, and the latest connector version that created the corresponding instances. We therefore maintain an additional named graph that aggregates triples from connector graphs that represent the latest versions. This enables querying current data while preserving complete history.

2.2.2.5 | *Rules.* Rule scripts (GitLab-versioned Python scripts) operate directly on the graph. They query relevant information, compute derived values, and write results back as new instances and relations. Versioning supports traceable updates to rule logic. One rule, for example, retrieves CFRP ply densities and geometries, computes laminate fiber volume fraction, and stores the result in the graph.

2.2.2.6 | *Data Availability.* In addition to the pipeline implementation published on GitLab, the resulting triples as well as the underlying experimental and simulation datasets (e.g., tensile test results and fracture analyses) are openly accessible via the MD data portal [18].

The FAIR data stored in the triplestore support three processing levels for statistical analysis and ML. For the digital part record, the user interface retrieves required information via SPARQL queries and visualizes it. For the remaining use cases, the system exports data to tabular formats (JSON/CSV) to enable statistical processing and ML workflows. Export uses a single master SPARQL query that returns all required variables as columns (e.g., tensile strength, overlap length, maximum curing temperature, and flags indicating experiment vs. simulation).

For use case 2, the system quantifies similarities and differences between selected batches (each batch corresponds to a subset of rows in the exported table). It uses Cohen's *d* for numerical variables and Jensen–Shannon divergence for categorical distributions. It ranks variables by their contribution to similarity or difference patterns.

For use case 3, the system performs conditioned correlation analysis. It groups samples by nearly identical configurations (e.g., same bolt diameter, overlap length, and curing temperature). It then computes Spearman correlations within each group and pools them using Fisher *z*-transformation. This controls for experimental-setup differences that confound global correlations.

For use cases 4 and 5, the system trains separate Random Forest regression models for each target variable (e.g., overlap length, tensile strength, and curing temperature). Models are served via an API and invoked by the user interface using user-provided inputs.

In this way, the workflow makes the data AI-ready in the sense outlined in the Section 1: the inputs to the statistical analyses and Random Forest models are semantically described, provenance-tracked, and queryable via SPARQL, allowing ML methods to consume traceable data with a reduced need for manual curation.

2.2.3 | *Application and Usage*

The user interface provides direct access to all use cases (Figure 4). For anomaly analysis, users label and compare subsets (e.g., a "good" batch vs. a "bad" batch). The system runs the statistical analysis described above and presents ranked results, for example, showing that the "bad" batch has significantly lower adhesive curing temperature. Users can click individual data points or search by specimen ID to open the digital part record, which displays available provenance and process information (e.g., curing-temperature cycle and surface-treatment type).

For correlation analysis, users select target variables (e.g., tensile strength). The interface shows ranked conditioned correlations and highlights influential factors such as adhesive tensile strength, CFRP–steel friction, or curing temperature.

For prediction, users choose a target (e.g., tensile strength), provide input parameters (e.g., overlap length and bolt diameter), and receive a prediction with an uncertainty interval. The

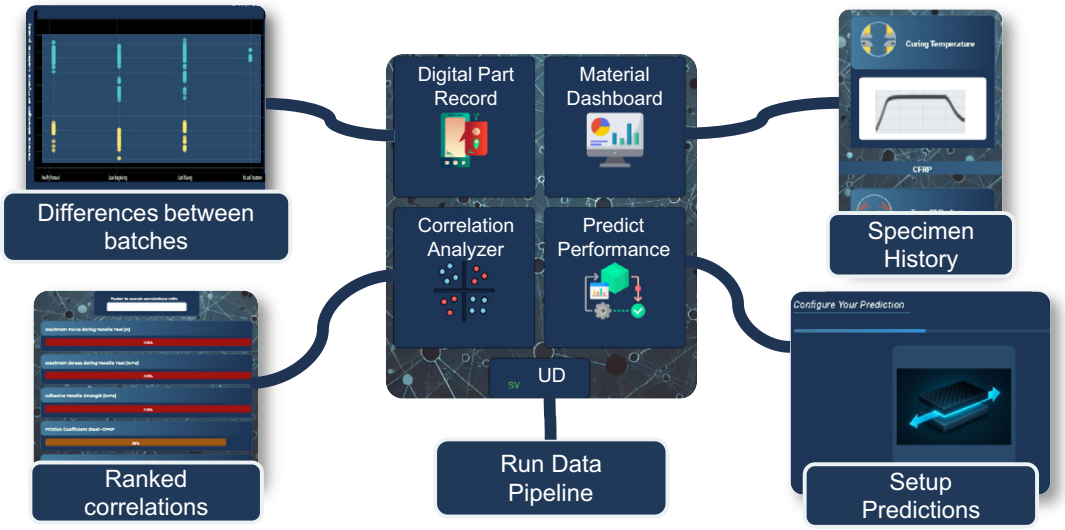


FIGURE 4 | Angular user interface for accessing the use cases.

interface also reports training and test set sizes, R^2 scores, and the raw training data used.

2.2.4 | *Limitations and Outlook*

Current matching between raw files and parsers and between parsers and connectors relies on keywords. This enforces strict naming conventions and scales poorly. Future work will replace keyword matching with more robust selection, for example, by computing semantic similarity between raw-file content and parser input requirements.

Statistics and ML components currently operate on exported tabular data. This requires a predefined master SPARQL query and an explicit export step. Future work will move analysis and learning closer to the graph by operating directly on the triple store, for example, via graph neural networks.

2.3 | DiStEL: Semantic Data Acquisition and Multiscale Simulation for Steel Components

MD Funding Phase 3; Project Started in 2025

The MD3 project DiStEL (**Di**gital material analysis along the entire value chain for **St**eel components; **E**fficiency, service **L**ife, and carbon footprint) addresses the digitalization of material data in an automotive value chain. Its overall goal is to provide a holistic view of the process, from raw material production through engineering, manufacturing, and use to the end of a vehicle's life and its dismantling. By making material data available and interpretable throughout the value chain, the project is intended to support holistic optimization with respect to key questions such as *R-X* strategies, product carbon footprints, and service life.

To enable this holistic view, two pain points regarding material data are addressed using workflow topics. The first concerns the availability of semantically annotated data stored in a triple store when only raw or unstructured data is available. The second pain point is a holistic simulation chain that includes data from different stakeholders in the value chain. The workflow tools developed in DiStEL will be made available through GitHub and linked from the project website: https://www.materialdigital.de/project/26.

To address the first pain point, a data acquisition pipeline (DAP) is being developed. The corresponding work package focuses on a pipeline that automatically or semiautomatically maps unstructured or raw project data to semantically annotated data stored in the triple store "OntoDocker." To achieve this goal, several mapping or transformation steps are implemented:

- Raw data are first converted into a JSON format using dedicated parsers.
- Second, the resulting JSON is transformed into a Canonical JSON (cJSON) representation.
- Subsequently, the cJSON is mapped to the PMDco ontology, resulting in a corresponding A-Box.
- Finally, the generated A-Box is stored in the OntoDocker.

To improve efficiency, the pipeline architecture supports parallel processing, enabling multiple tasks to run simultaneously. Containerized components are intended to support deployment at increasing workloads. Finally, the system is event-driven, reacting to incoming notifications. After successful testing, the resulting pipeline will be distributed through the GitHub organization of the PMD. More broadly, the DAP is intended to increase the availability of machine-readable, semantically annotated data and thereby support the AI readiness of material-specific value-chain data.

To address the availability of simulation data along the value chain, the second workflow work package is developing a fully integrated, multiscale simulation workflow that links component lifetime directly to processing conditions. This process is detailed in Figure 5. The central idea is to couple several specialized

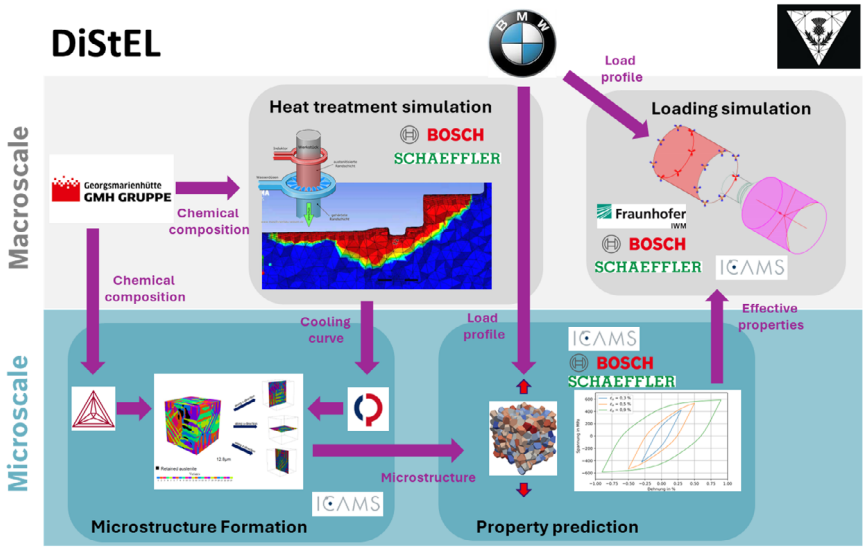


FIGURE 5 | Schematic depiction of the DiStEL simulation workflow.

simulation tools through consistent scale-bridging strategies so that information on the initial material state, the manufacturing route, the evolving microstructure, and the effective properties is propagated through a single, coherent workflow relying on pyiron.

The sequence below describes the target architecture. End-to-end execution of the integrated chain and validation of its scale transfers are not yet documented in this contribution. At the macroscale, component manufacturing simulations are carried out by the industrial partners. These simulations resolve the full processing history of the part, including cooling conditions.

The resulting spatially and temporally fields (e.g., temperature-time profiles) form the boundary conditions and input descriptors for the subsequent microstructure formation simulations performed in academia (ICAMS) on the microscale. In this step, the local processing histories are translated into realistic microstructural states, including phase fractions and morphology.

The microstructures are then passed within the workflow to microstructure property simulations at the microscale.

Here, phenomenological crystal plasticity models are employed to derive effective elasto-plastic properties, such as anisotropic yield surfaces. This step provides a crucial bridge between virtual microstructures and engineering-relevant material parameters. Finally, component-level fatigue lifetime predictions are performed based on the effective properties obtained from the microstructure property simulations. The loading profiles are thereby retrieved from real-world use cases from one of the industrial partners. This stage, jointly executed by academic and industrial partners, uses both local microstructural descriptors and component-scale stress and strain fields to estimate the overall service life of the component. In this way, the planned integrated DiStEL simulation workflow is intended to close the loop from initial material state and processing conditions to microstructure, from microstructure to mechanical response, and from mechanical response to lifetime. This integrated approach could provide a basis for process- and microstructure-informed lifetime design and systematic, simulation-driven optimization of industrial processing routes. Storing simulation results in pyiron nodes could also support future machine access. An agent-facing interface or agent-mediated execution of the integrated multiscale chain is not demonstrated here.

2.4 | DigitalModelling: Automated Workflow for Material Parameter Identification

MDFunding Phase 2

2.4.1 | Motivation

On the meso- and macroscale, modeling of microstructures and material properties aims to turn physical mechanisms into predictive tools that remain valid across conditions and applications. A decisive step is parameter identification: a well-posed, traceable calibration for sophisticated (or even simple) physical and constitutive models is required to be used with confidence [19]. For instance, the simulation of service cycles of high-temperature components (e.g., blades, disks, and rotors in turbines or aeroengines) requires advanced viscoplastic constitutive models within a finite element analysis (FEA) of the thermomechanical loading [20, 21]. These models typically involve a nontrivial set of model- and material-specific parameters that must be identified through calibration against appropriate experimental datasets, that is, by fitting the model response to measured stress–strain histories, creep curves, and/or cyclic hysteresis loops under well-defined loading conditions [22–24]. This procedure is usually cumbersome and prone to errors since it requires the use of several numerical tools and considerable manual data handling. The DigitalModelling project aims to develop a unified, reliable, and reproducible workflow for identifying material parameters of advanced constitutive models. The workflow integrates test-data processing, numerical simulation, parameter optimization, and model validation into a single automated pipeline. Its purpose is to calibrate material models by minimizing a weighted

least-squares objective function that quantifies the deviation between experiments and simulations and to assess predictive capability using independent validation data subsequently. Full automation is achieved by implementing the workflow within the pyiron environment, providing an integrated, scalable, and reproducible platform for all required tasks.

2.4.2 | *Workflow Structure*

The workflow starts with mechanical test data obtained from laboratory experiments. Depending on the constitutive model under study, these datasets may include tensile, creep, and low-cycle fatigue tests, as well as other specialized loading protocols designed to probe the material response under controlled deformation. While the experimental campaign has already been completed, a dedicated data-processing step is planned for future work to clean and prepare the raw signals by removing noise, correcting offsets, eliminating corrupted segments, and applying filtering or smoothing where necessary.

To interpret the experimental response, numerical simulations of the mechanical behavior of the material were performed under the same loading conditions as the physical tests. At present, the simulations are carried out using `Simul`, a BAM-developed tool to simulate homogeneous uniaxial and multiaxial tests. In future work, the framework will be extended to include additional commercial and open-source solvers, such as `OpenPhase`, `Abaqus`, `Ansys`, `Siemens NX`, and `CalculiX`.

Once the simulations are completed, the relevant numerical results, such as stress–strain curves, force–displacement responses, creep strain evolution, and cyclic hysteresis loops, are extracted and compared with the processed experimental data. Quantitative error measures are computed to describe the deviation between experiments and simulations. These errors are weighted and combined into a single objective function for subsequent optimization, providing a clear measure of how well the material model reproduces the observed response.

The next step is the automatic optimization of the material parameters. For complex constitutive models with a large number of parameters (typically between 5 and 20), manual parameter tuning quickly becomes inefficient and intractable. Instead, numerical optimization algorithms are used to systematically adjust the parameters. If reasonable initial estimates are available, local optimization methods such as the Nelder–Mead algorithm [25] or gradient-based approaches are recommended. If no reliable starting values exist, a global optimization strategy such as differential evolution [26] is used instead. Physically meaningful lower and upper bounds are defined for all parameters. Currently, optimization is performed using routines from the `SciPy` library [27]. In future developments, the framework will be expanded to include optimizers from other sources for more specialized calibration strategies.

After calibration, the identified parameters were validated using independent experimental data that were not part of the optimization process. These validation tests may involve different loading paths, strain rates, temperatures, or boundary conditions. The division of the data into training and validation data depends on the available testing data and must be performed by the user. The goal is to ensure that the model does more than simply fit the calibration data; it must also reliably predict new scenarios. Only after this validation step can the material parameters be considered suitable for further simulations and engineering applications.

2.4.3 | *Implementation of the Workflow Using pyiron*

To automatize and manage the entire parameter identification process, the workflow outlined above has been implemented within the `pyiron` [3, 28] platform, as shown in Figure 6.

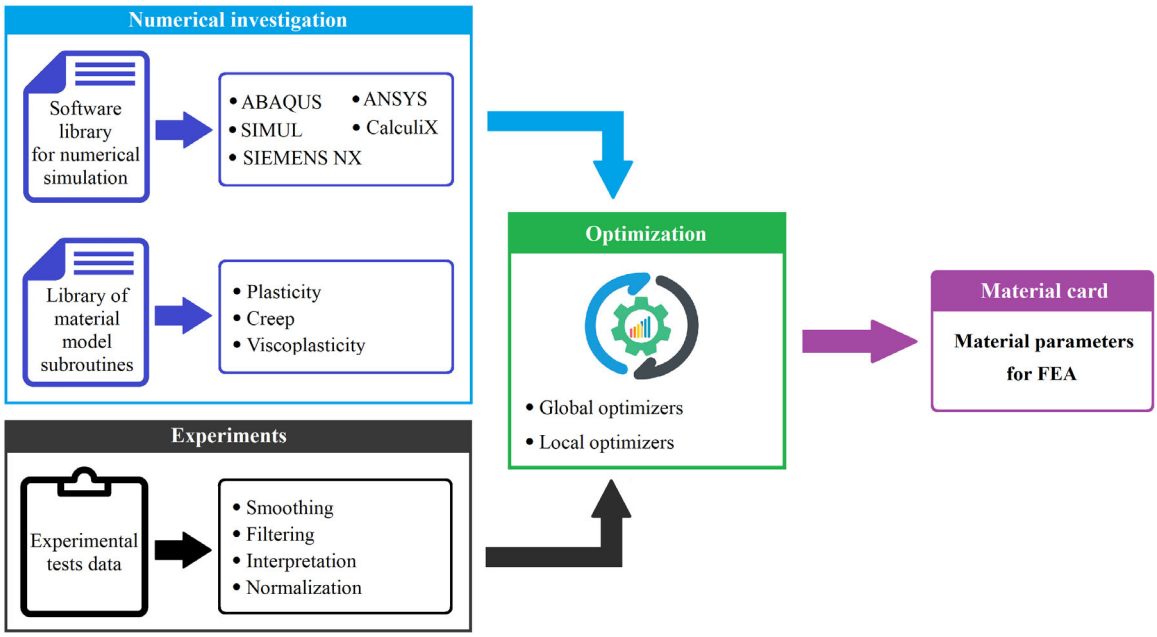


FIGURE 6 | Schematics of the workflow for model parameter determination.

pyiron provides a unified environment that links experimental data, numerical solvers, optimization algorithms, and visualization tools in a consistent and reproducible way. The framework supports local data sources and can be expanded to include external ones, allowing flexible integration of solvers, data-processing routines, and optimization methods.

Each simulation job, including its inputs, outputs, logs, and metadata, is automatically stored. This structured data management supports traceability and reproducibility across the workflow steps, including individual optimization iterations and validation runs.

A user-friendly graphical interface was available for configuring simulations, handling material parameters, and inspecting both experimental and numerical results. This reduces the need for manual file editing and allows users to easily monitor the optimization progress and validation performance.

pyiron supports execution on both local systems and HPC clusters, allowing the workflow to scale to large parameter studies or computationally intensive simulations. Furthermore, its modular architecture enables the use of multiple solvers and the integration of different libraries implementing the material models, such as UMAT for Abaqus or USERMAT for Ansys, ensuring compatibility with a wide variety of constitutive models and computational requirements. The code and data files related to Digital Modelling are available at https://github.com/ehsanfarahani62-gif/DigitalModelling.

2.4.4 | *Application to Creep Testing*

To illustrate the practical use of the proposed workflow, it is applied to the identification of creep material parameters for the nickel-based superalloy IN718 at 600 °C. A set of creep experiments was performed under different stress levels [29], and the corresponding numerical simulations were performed under identical loading and boundary conditions. The material parameters were then refined through an automated optimization process to minimize the difference between the experimental and simulated creep responses. The resulting agreement between experiments and simulations is presented in Figure 7.

2.4.5 | *Ontology-Based KG for FAIR Material Cards (Workflow Infrastructure)*

To ensure that model calibration results can be reused with confidence across projects, DigitalModelling complements the automation in pyiron with an ontology-based knowledge-graph infrastructure. The goal is to integrate heterogeneous information sources—experimental datasets, simulation settings, model descriptions, optimization results, and provenance metadata—into a coherent, queryable representation that supports traceability and FAIR-aligned reuse. In this setup, the KG acts as the semantic backbone between local material databases and the modeling workflow, and it provides the basis for generating consistent material cards.

Figure 8 illustrates the overall architecture. Local databases (e.g., creep, fatigue, and microstructure) are connected to data-analysis steps and mapped into a shared graph representation. In parallel, a model library (model description, executable/source code, modular generators, and benchmark results) is represented as structured information that can be linked to specific materials, datasets, and calibration contexts. Workflow executions then read inputs from the graph and write back results as structured outputs, enabling downstream material-card generation while preserving provenance and version information.

A key aspect is the explicit semantic connection between the calibrated parameter set and its evidential basis. Calibrated parameters are stored as workflow outputs and linked to the

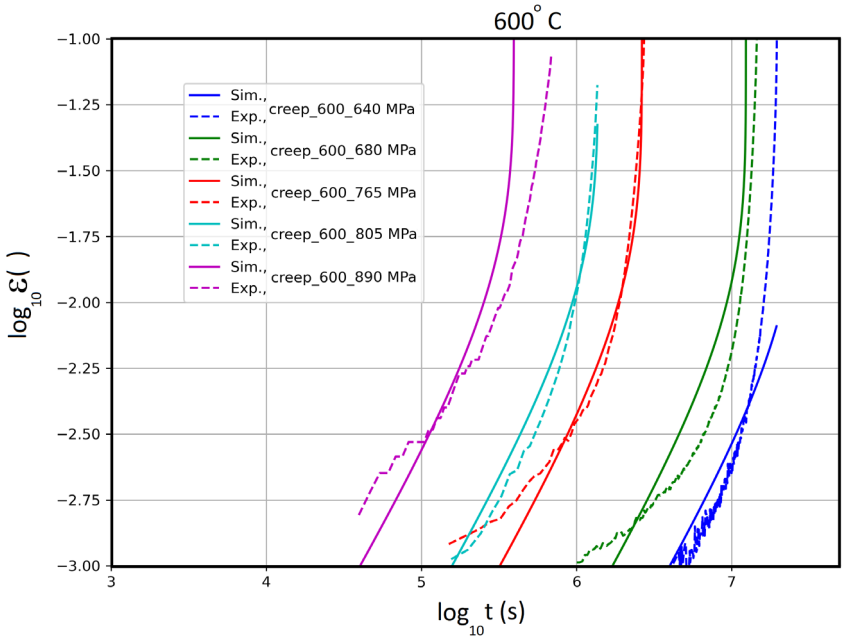


FIGURE 7 | Experimental creep results and optimized numerical simulations, showing the best agreement used to determine the material parameters.

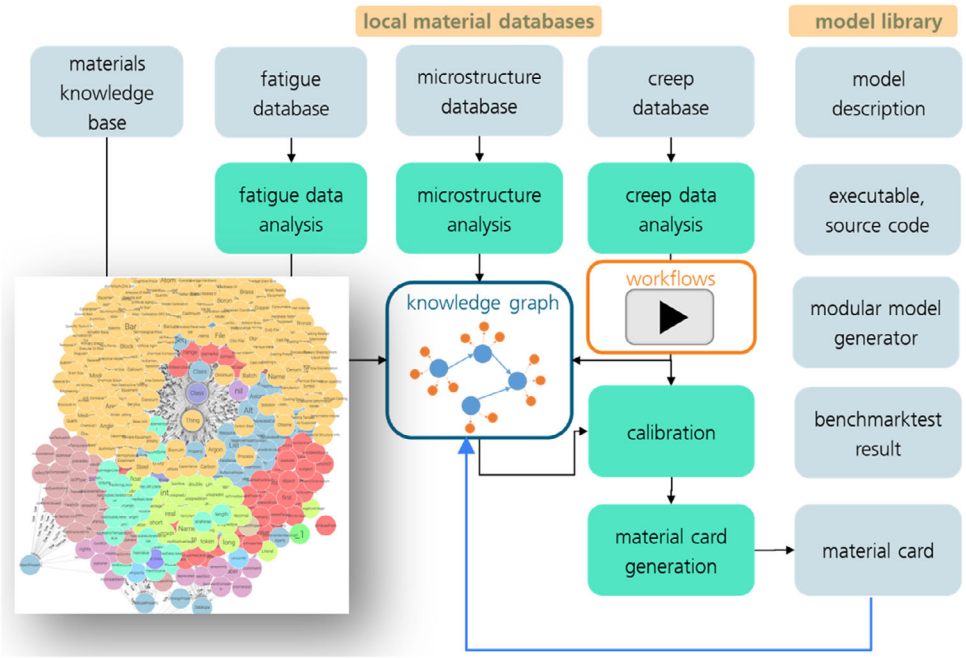


FIGURE 8 | Overview of ontology-based knowledge graphs in DigitalModelling, linking local databases and a model library with executable workflows to enable calibration traceability and material-card generation.

underlying experimental datasets, the model version and solver configuration, and the relevant material identifiers (including batch and supply-chain context where available). This allows cross-domain queries that are difficult to answer in file-based workflows, such as retrieving all calibrated parameter sets for a given material batch, tracing which experiments were used for calibration versus validation, or identifying which workflow version produced a parameter set that is used in subsequent simulations.

The resulting material card is generated from the KG as a structured digital object with stable identifiers, machine-readable content, and explicit provenance links. This supports both engineering reuse (e.g., selecting parameters for a new simulation scenario) and auditability (e.g., tracing reported values back to datasets and workflow artifacts). The same infrastructure can be aligned with Digital Product Passport (DPP) principles by providing a consistent schema for identifiers, process context, calibrated parameters, and linked evidence.

2.4.6 | Material Card Pass to Enhance the Traceability of Calibrated Material Cards

Within DigitalModelling, the generated material card is treated as a digital asset that can be accessed both by users and by automated tools. To serve both purposes, a DPP-style interface is used to expose the material card content directly from the KG via identifier-based queries. The interface provides a human-readable structure for navigation, while the underlying representation remains machine-interpretable and consistent with the ontology-driven schema.

An example for the IN718 creep use case is shown in Figure 9. The information is organized into dedicated views covering provenance, material information, software and equipment, input and control parameters, and calibrated parameters. A central element is the explicit handling of linked artifacts ("Linked Files"), which connects the material card to the datasets and documents that support it (e.g., experimental creep data, dataset letters,

Digital Product Passport for: MaterialCardsdChaboX_IN718_BAM_v1

Provenance | Material | Software and Equipment | Input and Control Parameters | Calibrated Parameters | **Linked Files**

process_label	file_label	file_role	file_name	file_URL
CalibrationsdChaboXModel__a	DocumentationFile__49	input	creep_tests_in718_mpa-stuttgart	creep_tests_in718_mpa-stuttgart
CalibrationsdChaboXModel__a	DataSetLetter__52	input	datasetletter_schabox-calibration_in718_600c.docx	datasetletter_schabox-calibration_in718_600c.docx
CalibrationsdChaboXModel__a	ModelLetter__46	input	modellsteckbrief-sdchabox-en-2025-05-23.pdf	modellsteckbrief-sdchabox-en-2025-05-23.pdf
CalibrationsdChaboXModel__a	MaterialCard__23	output	MaterialCardsdChaboX_IN718_BAM_v1	MaterialCardsdChaboX_IN718_BAM_v1
CalibrationsdChaboXModel__a	ImageFile__26	output	creep_600-loglin.png	creep_600-loglin.png

FIGURE 9 | Example of a DPP-style material card interface for a DigitalModelling digital asset, including a tabular view of linked files that provide evidence and traceability.

model documentation, workflow outputs, and generated figures). This helps ensure that reported parameter values remain connected to their calibration inputs, assumptions, and evidence rather than becoming detached results.

By storing the material card and its evidence links in the same semantic infrastructure, updates can be handled in a controlled way: new data or model versions could trigger recalibration, producing a new parameter set and an updated material card while preserving provenance and version relationships. This would make the material card a maintainable digital asset that supports traceable reuse of calibrated constitutive models in subsequent engineering workflows.

2.5 | DiMad: Workflow-Oriented Parameter Studies With MICRESS

MD Funding Phase 2; Updated Contribution

Within the MD project DiMad, numerical simulations are employed to predict microstructure formation during wire-based additive manufacturing processes. In particular, phase-field simulations using the software MICRESS [30] are used to investigate how process parameters, such as alloy composition or thermal boundary conditions, influence microstructural evolution. A typical example is the systematic analysis of solidification conditions to establish process–microstructure correlations, where quantities such as phase selection and dendrite tip temperature are evaluated over a range of processing parameters (Figure 10).

Such investigations require parameter studies in which numerous simulations with systematically varied input parameters are generated, executed, and evaluated. While conceptually straightforward, these studies become labor-intensive when performed manually and often rely on project-specific scripting solutions that are difficult to reproduce, reuse, and maintain.

Workflow management systems (WMS) provide a structured approach to automating these tasks while improving reproducibility and portability. Integrating legacy simulation software such as MICRESS into modern workflow environments, however, requires bridging application-specific file formats and execution models with the abstractions expected by workflow systems.

This contribution presents a workflow-oriented integration of MICRESS for automated parameter studies. The approach is demonstrated using a representative MICRESS benchmark and implemented in two WMS, pyiron [3] and SimStack [2]. These frameworks were selected because they represent the primary workflow solutions currently established within the PMD ecosystem. The following sections describe the workflow design, its implementation in both systems, and a comparison of the resulting approaches.

2.5.1 | *Use Case*

Using full-scale DiMad simulation scenarios to evaluate workflow concepts is impractical as individual MICRESS simulations can be computationally expensive and slow to iterate. To decouple workflow design from runtime considerations, a simplified yet representative use case was defined.

An official MICRESS benchmark simulating early-stage microstructure growth from the liquid state was selected. Such benchmarks are routinely used during MICRESS development to validate model consistency and ensure reproducibility after code modifications. Although simplified, the benchmark encompasses all essential steps of a realistic simulation workflow: input generation, solver execution, and output analysis.

From a physical perspective, the benchmark describes the early stage of a crystalline nucleus during solidification from the melt using a phase-field formulation. At this stage, microstructural

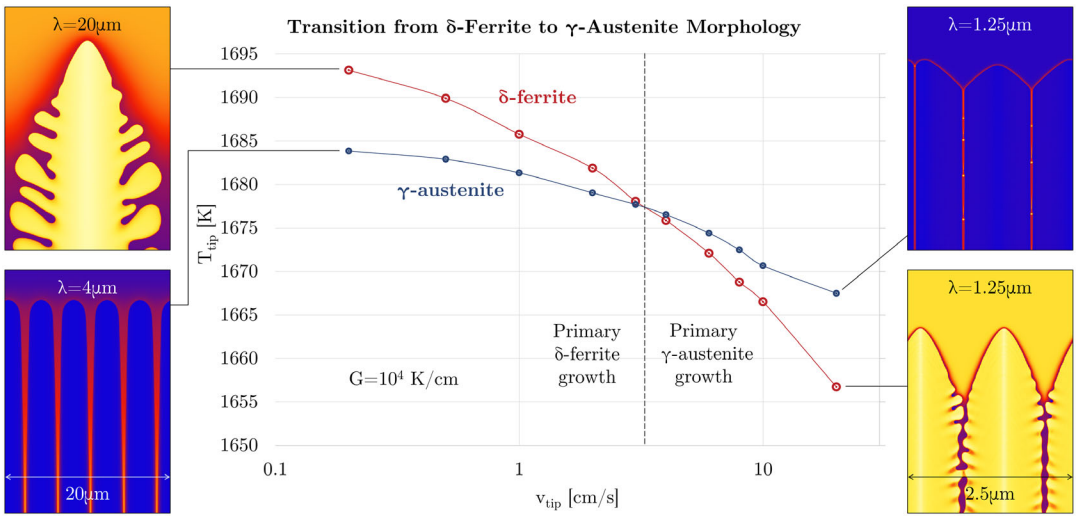


FIGURE 10 | Representative parameter study performed within the DiMad project to establish a process–microstructure correlation for the austenitic steel 316L under rapid solidification conditions. Individual MICRESS simulations are performed for different stationary tip velocities at a fixed thermal gradient of $G = 10^4$ K/cm. The resulting tip temperatures for primary δ-ferrite and γ-austenite growth are evaluated, while representative phase-field snapshots of the Mo distribution illustrate the corresponding steady-state microstructures. Such studies require the repeated generation of simulation inputs, execution of MICRESS, and postprocessing of the results, motivating workflow-based automation.

development is dominated by the interplay between curvature-dependent interface forces and thermodynamic bulk driving forces. The evolving morphology can therefore be characterized by a small number of scalar control parameters, such as interface energy or thermal undercooling. An analytical reference solution based on the Turnbull criterion is available, enabling automated validation of numerical results and regression testing within workflow-driven studies.

From a computational standpoint, each simulation run represents an independent realization of the same physical model with a modified set of input parameters. No data dependencies exist between individual runs, rendering the benchmark inherently parallel and well-suited for parameter studies orchestrated by WMS.

The selected benchmark offers several advantages:

- Extremely short runtime (on the order of seconds per simulation),
- Physically meaningful and analytically verifiable results,
- Structural similarity to production-scale MICRESS simulations,
- Independence of individual simulation runs.

These properties make the benchmark particularly suitable for investigating workflow-related aspects such as parameter variation, data handling, validation, and reproducibility while deliberately neglecting runtime scalability issues.

2.5.2 | *Workflow Abstraction*

Both pyiron and SimStack are Python-based WMS that expect simulation codes to expose well-defined input and output interfaces. MICRESS, however, is a standalone executable operating on application-specific file formats that are not natively supported by either system. Consequently, explicit adapters were required to integrate MICRESS into both workflow environments.

Before discussing the system-specific implementations, the MICRESS benchmark workflow can be abstracted into a sequence of logically independent stages, independent of any particular workflow technology:

1. **Parameter definition**: Specification of physically meaningful input parameters and their variation ranges.
2. **Input materialization**: Translation of abstract parameter sets into concrete MICRESS Driving Files.
3. **Simulation execution**: Invocation of the MICRESS solver as an external executable.
4. **Postprocessing and validation**: Extraction of relevant observables and comparison with analytical expectations.

This abstraction separates physical modeling concerns from workflow orchestration and provides a common conceptual basis for both implementations.

2.5.3 | *Input Handling*

MICRESS simulations are configured via so-called Driving Files (`.dri`), which resemble the transcript of an interactive question-and-answer session. The structure of these files is highly MICRESS-specific and not governed by a simple grammar; depending on previous responses, additional questions may appear. While this complicates generic input generation, parameter studies typically modify only a small, fixed subset of parameters. Under this assumption, input generation becomes deterministic.

In the DiMad workflow, this was addressed by templating an existing validated benchmark Driving File using the Jinja templating framework. Parameters selected for variation were explicitly annotated and automatically substituted, enabling consistent generation of MICRESS input files for each parameter set. This approach relies only on the text-based nature of the configuration files and is therefore transferable to many legacy simulation codes with similar input mechanisms.

2.5.4 | *Output Handling*

MICRESS produces several nonstandard output formats. Time-dependent scalar quantities are written to CSV-like text files with multiline comment headers, while spatially resolved field data are stored in proprietary compressed binary formats.

To bridge these formats into the Python ecosystem, the open-source MicPy library was employed, which provides reusable Python interfaces for MICRESS output data [31]. MicPy converts MICRESS tabular output files into Pandas DataFrames with correctly interpreted headers and provides access to binary field data as NumPy arrays. By translating application-specific outputs into standard Python data structures, postprocessing, validation, and visualization could be implemented uniformly across both workflow systems (Figure 11).

2.5.5 | *Workflow Implementation and Comparison*

While the underlying workflow logic is identical in both systems, pyiron and SimStack expose this logic through distinct interaction paradigms that lead to different strengths and tradeoffs.

pyiron follows a script-based, Python-native approach that integrates naturally with interactive environments such as Jupyter notebooks. Workflow logic, parameter variation, and postprocessing are expressed explicitly in Python code, closely mirroring established batch-processing concepts such as working directories, input files, and output artifacts. This design enables rapid prototyping, transparent debugging, and tight integration with existing Python-based analysis tools.

SimStack, in contrast, follows a client-server architecture with a graphical user interface. Workflow logic is encapsulated in so-called Workflow Active Nodes (WaNos), which define input widgets, execution commands, and data flow. This approach emphasizes separation between workflow authors and workflow users: once a WaNo is defined, parameter studies can be executed through the GUI without requiring detailed knowledge of the underlying implementation. At the same time, authoring

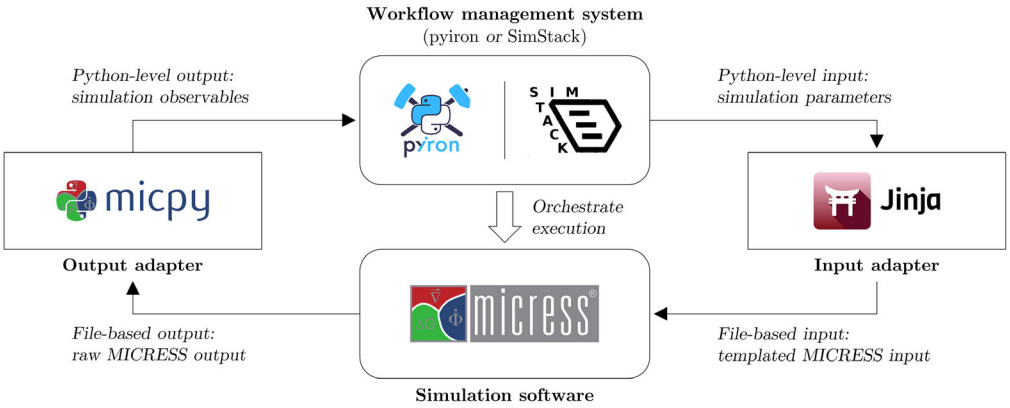


FIGURE 11 | A legacy, file-based MICRESS executable is orchestrated by a workflow management system via explicit input and output adapters. Text-based input files are generated using Jinja templating, while simulation outputs are converted into Python data structures using MicPy, enabling reproducible parameter studies without modifying the simulation code.

WaNos involves a higher initial effort and requires familiarity with SimStack-specific conventions.

From a practical perspective, pyiron proved particularly well suited for workflow development and exploratory studies by Python-literate users, whereas SimStack offers advantages for deploying predefined workflows to nonexpert users in a guided and reproducible manner. Both systems were capable of executing MICRESS simulations reliably once the required adapters were implemented.

2.5.6 | Conclusion, Lessons Learned, and AI-Readiness

This work demonstrates that modern WMS can be pragmatically integrated with a legacy file-based simulation code such as MICRESS using a small set of explicit input and output adapters. The combination of text-based input templating and Python-based output parsing proved sufficient to enable reproducible and automated parameter studies without modifying the simulation code itself.

From a user perspective, pyiron is particularly attractive for MICRESS users with Python experience who value transparency, rapid prototyping, and tight integration with existing analysis workflows. SimStack, while requiring a higher initial setup effort, enables encapsulation of simulation expertise into reusable workflow components that can be executed through a graphical interface by nonexpert users.

Beyond these practical advantages, the presented workflow abstraction also provides a basis for future AI-assisted simulation workflows. Native MICRESS Driving Files represent a context-dependent interface whose valid structure depends on previous inputs, making direct generation of valid input files difficult for both conventional automation and AI systems. By replacing this implicit interaction with parameterized templates and explicit workflow interfaces, the integration exposes a stable set of physical input parameters while hiding the application-specific complexity of the underlying file format. Such structured interfaces constitute an important prerequisite for integrating legacy simulation codes into AI-supported research workflows.

To support reuse and reproducibility, the workflows developed in this study were published openly and registered in the MD Workflow Store. The pyiron-based MICRESS workflow, including two alternative notebook implementations of the same parameter study logic, is available as a single documented repository and corresponding Workflow Store entry [32]. The SimStack-based workflow is provided as a documented workflow configuration registered in the Workflow Store [33]. In addition, the two SimStack WaNos used for MICRESS execution and postprocessing were published as standalone GitHub repositories to facilitate reuse in other SimStack workflows [34, 35].

2.6 | AnAttAl: pyiron-Based Workflow for Resistance Spot Welding (RSW) Simulation

MD Funding Phase 2

2.6.1 | Motivation and Goals

RSW simulations typically involve a large number of heterogeneous and tightly coupled tasks, including the preparation of geometric models for electrodes and sheets, the definition of material properties and contact parameters, mesh generation, and the application of experimentally measured process parameters such as current, voltage, and electrode force. In addition, the generation and execution of Abaqus input files as well as the extraction and postprocessing of results from ODB files are required.

In practice, these steps are often carried out manually or implemented through isolated scripts. Such fragmented workflows are prone to errors, difficult to reuse, and lack transparency and traceability. Although the proposed workflow could also be implemented using standalone Python scripts combined with conventional workflow tools, pyiron was selected for the following three main reasons:

- **Experiment–Simulation Coupling and Simulation Automation:** Seamless integration of experimental process parameters, Abaqus preprocessing, external Fortran calculations, and automated simulation execution within a unified workflow.

- **Workflow Data Management and Reproducibility:** Automatic recording of workflow parameters, input data, intermediate results, and final simulation outputs, ensuring complete traceability, standardized data management, and reproducibility.
- **Unified Workflow Environment:** Centralized orchestration of the complete simulation workflow, reducing manual configuration effort and improving usability, extensibility, and reproducibility compared with isolated scripting approaches.

Based on these capabilities, the proposed approach enables GUI-supported execution of complete RSW simulations, ranging from user-defined input parameters to final ODB-based results, within a single coherent environment.

2.6.2 | *Software Integration*

pyiron serves as the central framework of the proposed RSW simulation workflow by coordinating the interaction between the commercial finite element (FE) software Abaqus and an external Fortran-based solver. As illustrated in Figure 12, the workflow integrates all preprocessing steps within Abaqus, including geometric modeling, material parameter definition, contact modeling, and mesh generation, under the unified orchestration of pyiron.

Abaqus is primarily used to generate discretized geometric and contact information, such as nodes, elements, and contact topology. This mesh-based information is subsequently exported and provided as input to the Fortran-based electrothermal solver, where the physical calculations are performed. Welding process parameters, including current, voltage, and electrode force, were obtained from external experimental or process data sources and were directly imported into the Fortran solver during the simulation.

Throughout the workflow, pyiron manages the automated execution of the individual simulation stages, including the invocation of external programs, the control of job sequencing, and the consistent handling of working directories and file paths. This orchestration enables a seamless coupling between the mesh and contact data generated in Abaqus and the physics-based computations carried out in the Fortran solver while minimizing manual intervention and configuration effort.

In addition, the workflow provides a structured framework for managing workflow parameters, simulation inputs, and simulation outputs, thereby supporting reproducibility and consistent data organization.

The final simulation results are generated in the form of Abaqus ODB files, which are used for subsequent postprocessing and visualization. By encapsulating all simulation stages within a single execution environment, the proposed workflow enables fully automated, end-to-end RSW simulations and provides a robust and extensible foundation for future developments. The developed workflow is publicly available through the MD Workflow Store: https://workflows.materialdigital.de/workflow/AnAttAl_MPA.

2.6.3 | *Challenges*

One of the challenges encountered during the development of the workflow is the absence of native job classes for Abaqus or Fortran-based solvers in pyiron. As a result, the execution of these tools required custom wrapper implementations based on Windows command-line calls, including dedicated mechanisms for error handling and logging.

Ensuring reproducibility and stable execution within a Jupyter Lab environment posed additional difficulties. Running Abaqus from within this environment required specific configuration steps to guarantee consistent path handling, persistent storage of user-defined parameters, and reliable interaction with external executables.

Finally, certain parts of the workflow still rely on commercial software solutions, such as FAMOS, for processing welding machine data. While these tools are currently necessary to ensure compatibility with existing industrial data formats, they represent a limitation with respect to full open-source integration.

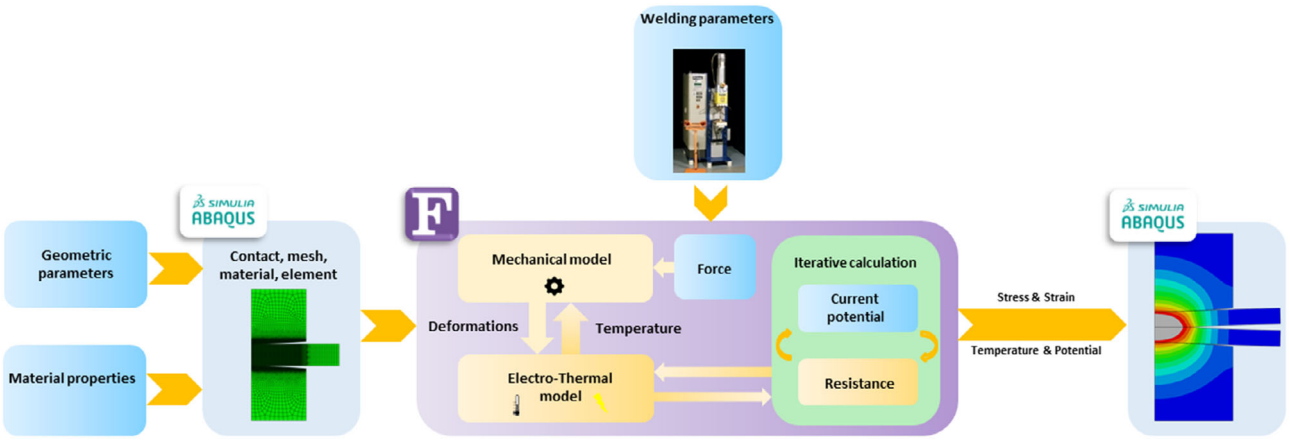


FIGURE 12 | Resistance spot welding simulation workflow integrated via pyiron, combining the finite element software Abaqus and an external Fortran-based solver.

2.6.4 | *Conclusion*

The pyiron-based workflow presented in this work integrates the individual steps of RSW simulation into a unified and automated framework. By combining geometry definition and mesh generation in Abaqus with external electrothermal calculations implemented in Fortran, the workflow reduces manual intervention and improves the reproducibility and transparency of the overall simulation process.

pyiron acts as the central orchestration layer, managing the execution of external programs, data flow, and file handling throughout the workflow. The final simulation results are generated in the form of Abaqus ODB files, which enable standardized postprocessing and visualization using established analysis tools. The modular structure of the workflow facilitates systematic parameter studies and provides a flexible basis for further methodological extensions.

Although the current workflow does not directly implement artificial intelligence or machine-learning algorithms, it establishes the necessary data infrastructure for future AI-assisted simulation workflows through the systematic generation and management of simulation data.

- **AI-ready data pipeline:** The workflow automatically generates structured, machine-readable, and traceable data throughout the complete simulation process, including input parameters, intermediate workflow data, and final simulation results.
- **Integration of experimental and simulation data:** Experimentally measured welding parameters, such as current, voltage, and electrode force, are incorporated as simulation inputs and can be associated with the corresponding simulation outputs within the unified workflow. This provides a consistent basis for the construction of combined experiment–simulation datasets.
- **Foundation for future machine-learning applications:** The structured and traceable datasets generated by the workflow can be systematically reused as training and validation data for future machine-learning models, surrogate models, and data-driven process optimization approaches.

Based on this data-oriented workflow architecture, future developments may include the integration of additional physical models, accelerated parameter exploration using surrogate models, and data-driven analysis and optimization of aluminum RSW processes.

2.7 | OntOMat: Ontology-Based pyiron Workflow for Multiscale FE Simulation

MD Funding Phase 2

2.7.1 | *Motivation and Goals*

A common challenge within fiber-reinforced composites is to deduce the effective thermo-mechanical properties of a composite material based on its constituents. Widely used analytical approaches include the Mori–Tanaka homogenization [36], the variational Hashin–Shtrikman bounds [37, 38], or the semi-empirical Halpin–Tsai approach [39]. Although analytical approaches offer fast results, they are limited by the fact that the microstructure of a composite can only be described on a statistical average, for example, using fiber orientation tensors [40] or average fiber lengths, which is where the term *mean-field* homogenization comes from. To consider complex microstructures, the homogenized properties are usually calculated using *full-field* methods, such as FE methods. It is computationally not possible to consider the full microstructure, for example, each fiber, in a simulation on a component level. To alleviate this issue, a multiscale simulation is performed. Starting from the microstructure, for which information of the spatial configuration and the mechanical properties of the constituents is required, the homogenized properties can be used in subsequent larger scales, that is, meso and macroscales. Here, homogenized properties describe a set of homogeneous thermo-elastic properties which effectively reproduce the physical response for a given domain for a given load. In other words, the deviation in the thermo-elastic response between the use of homogenized parameters versus locally inhomogeneous lower-scale parameters is minimized. A multiscale simulation makes it possible to investigate how changes in material or geometric properties at the microscale affect the effective properties at the macroscale.

On one hand, OntOMat is concerned with semantically describing these processes. On the other hand, the KG built upon the ontology in OntOMat contains information that can be used in numerical processes, such as the multiscale simulation, to further expand the information pool. Based on material properties of the constituents, a workflow can be designed to automatically perform a multiscale simulation of a desired composite. Therefore, an interaction or API with the KG is necessary, which first retrieves the available material information for the microstructure constituents, then performs script-based multiscale simulations, and feeds back generated information to the KG, for example, effective stiffness components on different scales. This allows the semantic relationships of the ontology of processes and discrete information already contained in the KG to be used to generate dependent information or, for example, to perform parametric studies on input materials or microstructure variations.

2.7.2 | *OntOMat Ontology (OMO) and KG*

One project goal of OntOMat is the creation of an ontology for (a) fiber-reinforced materials and their characteristics, (b) related manufacturing and characterization processes of these materials, and (c) (multiscale) simulation processes to determine certain characteristics or behavior. The OMO is conceptually split into three tiers, where the top level is alternatively the industrial data ontology (IDO) [41] or PMDCore 3.0 ontology [42]. The middle level is a handcrafted structure developed with support from domain experts, and the lowest level is automatically generated based on two industry standards concerning material classes and process descriptions. An important part comprises material classes and material qualities, that is, material properties, which are then used to define material products based on these material classes. Another important part describes the characterization, manufacturing, and simulation processes from an “a posteriori”

perspective; thus, the process path (e.g., based on observations stored in logs) of the respective process is captured, and (currently) not the process plan modeled "a priori." The above patterns lead to three main components:

- Physical products, basic material classes, and material property classes, where a wide range of (behavioral) material properties specific to composites are introduced;
- Classification of materials that represent the taxonomy of the IEC 62474:2018 standard, and automatic classification of composite materials according to the structure of their base materials;
- Process models are based on classes and properties that capture the VDI/VDE 3682 Formal Process Language (VDI 3682 language), which is a visual modeling language used in engineering/industry applications.

The ontology is published in the MD GitHub repository at https://github.com/materialdigital/ontomat-ontology.

Figure 13 shows the integration of a pyiron workflow and a KG, including the population of KG instance data based on Excel templates and the output of the VDI process modeling tool. Note that the population of KG instances is not part of the workflow and happens before by several ETL pipelines based on Reasonable Ontology Templates [43].

2.7.3 | *Multiscale Homogenization*

To demonstrate how the developments in OntOMat can be used in a multiscale simulation workflow, a two-step homogenization scheme with minimal user input was chosen. This includes the microstructural homogenization on constituent level and a subsequent meso-structural homogenization on laminate level. The FE solver used is `Abaqus` but generally should not be limited to a specific FE environment. The individual simulation steps are described below.

The microstructural simulation is based on a periodic representative volume element (RVE) of a unidirectionally (UD) fiber-reinforced composite with a hexagonal fiber packing arrangement, as is illustrated in Figure 14.

The user input is limited to selecting both materials of the constituents, that is, fiber and matrix material, via unique Internationalized Resource Identifiers (IRIs). It needs to be stressed that the user does not provide the material parameters themselves, which are necessary for the simulation. Additionally,

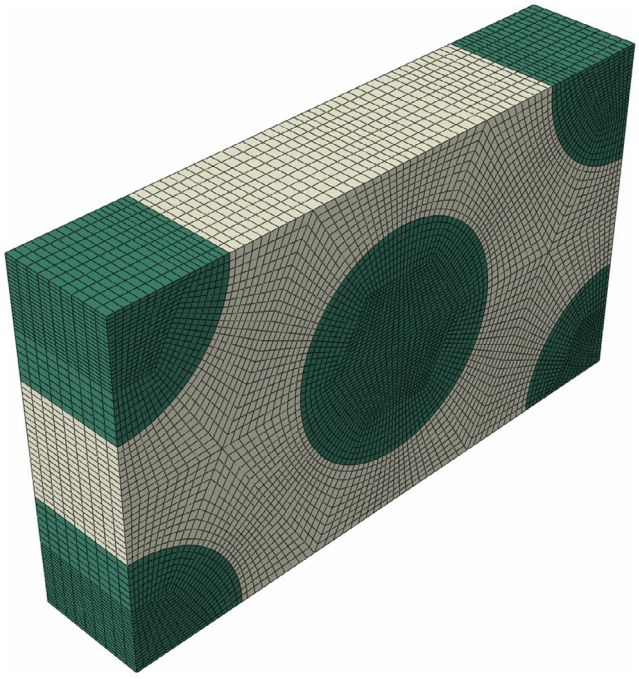


FIGURE 14 | Representative volume element of a periodic microstructure of a unidirectional fiber-reinforced composite. The green circular shapes are the fibers, while the remaining white volume is the matrix material.

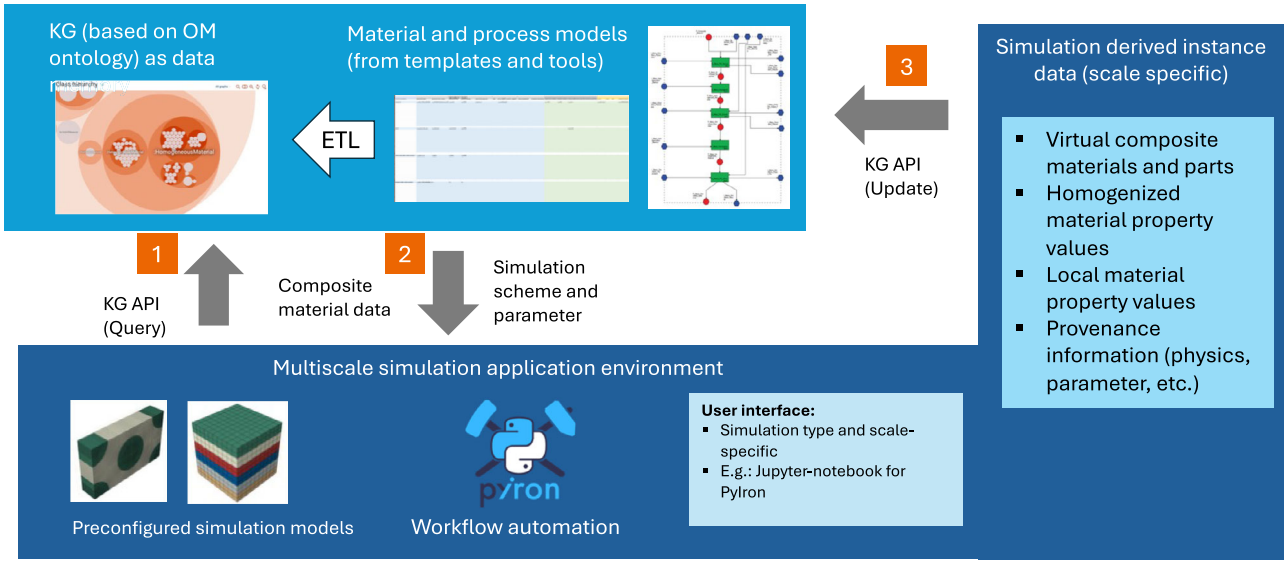


FIGURE 13 | Concept of KG and workflow integration.

the user determines the geometric properties, which are the fiber radius r_f, the depth of the RVE and the fiber volume fraction v_f. Following a query to the OntOMat KG through a `Python` API to retrieve the elastic material parameters of the constituents, the FE file was generated based on the `Python` scripting interface of `Abaqus`. The generation of the microstructure and the regular mesh is computed via the `Abaqus` Micromechanics Plugin [44]. Additionally, the plugin defines the boundary conditions for the six loading scenarios to cover all possible 21 elastic constants, which concludes the preprocessing of the homogenization simulation for the microstructure.

The simulation itself is performed using the `Abaqus` linear perturbation solver via a Bash script on a High Performance Computing (HPC) cluster, which operates on a SLURM management system. Once the simulation concludes, a postprocessing script is called via the `Abaqus` scripting interface to extract all 21 independent homogenized elastic stiffness components of the RVE, that is, the stiffness matrix C^{micro}. The effective stiffness C^{micro} is then passed back to the OntOMat KG and the subsequent meso-structural simulation.

Like the microstructural simulation, the meso-structural simulation is based on a periodic RVE of a laminate consisting of n laminas or plies, respectively. Each lamella consists of the same UD material that was homogenized in the microstructural simulation, only with different ply angles ψ_i. An example of a meso-structural RVE is given in Figure 15.

The user input determines the number of plies n and their corresponding angles ψ_i. Each ply thickness is identical. Based on the user input, the preprocessing is performed through the `Python` scripting interface of `Abaqus` to construct the FE file of the required RVE. The elastic material parameters are taken from the preceding microhomogenization.

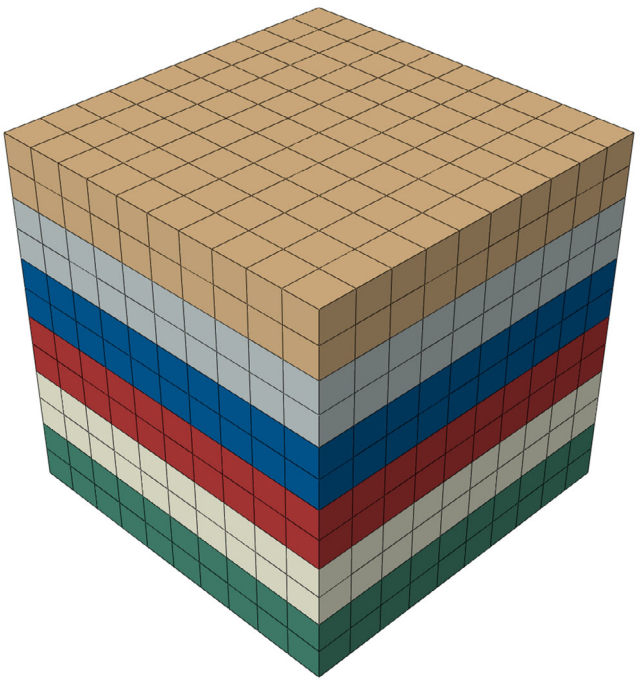


FIGURE 15 | Representative volume element of a periodic meso structure of a laminate consisting of $n = 6$ laminas. Each lamina is made from the same UD fiber-reinforced composite, only with different ply angles ψ_i.

After the preprocessing, the subsequent steps are identical to the microstructural simulation. The simulation is performed using the `Abaqus` linear perturbation solver via a Bash script and the postprocessing uses the same script as before to extract all 21 elastic stiffness components, that is, C^{meso}. The effective stiffness C^{meso} is then passed back to the OntOMat KG.

2.7.4 | *Workflow Structure*

The backbone of the multiscale homogenization workflow is `pyiron`. The key requirements are as following:

- the workflow needs to take care of managing the required simulation directories,
- the user input should be limited to a minimum,
- the workflow needs to be capable of coordinating (proprietary) FEM software, for example, `Abaqus`,
- workflow modules or nodes should be reused when possible, that is, compositions should be utilized.

The process structure within the ontology in OntOMat is based on compositions. For example, the *generic FE simulation* is repeatedly found in a concretized way in the *multiscale FE simulation*. This gives particular weight to the last point in the list of requirements above.

The final workflow structure is depicted in Figure 16.

The development environment used is a `Jupyter Notebook` in which the individual `pyiron` nodes are defined. In the following, the corresponding task of each node is summarized:

- **Initialize**: Generation of necessary (temporary) directories in which simulation files can be generated and FE results can be saved and evaluated.
- **OntOMat Query**: API call to retrieve the elastic material parameters based on a SPARQL query.
- **Preprocess Micro**: Script-based generation of FE input files, that is., the RVE, for the microsimulation.
- **FEM Solver**: Call to the SLURM manager on the HPC to start the FE solver.
- **FEM Evaluator**: Script-based evaluation of the FE results and extraction of the effective stiffness components.
- **OntOMat Upload**: API call to store generated results in the OntOMat KG.
- **Finalize**: Clean up of the generated directories.

Several tasks of the workflow need to be performed in a proprietary FEM software, in this case `Abaqus`, which requires tasks to be sent outside the `Jupyter Notebook` environment. This includes the generation of the RVEs, the starting of the FE solver, and the evaluation of the results. All of this was performed using the `subprocess` library in `Python`. It needs to be highlighted that the workflow above successfully reused several of the implemented

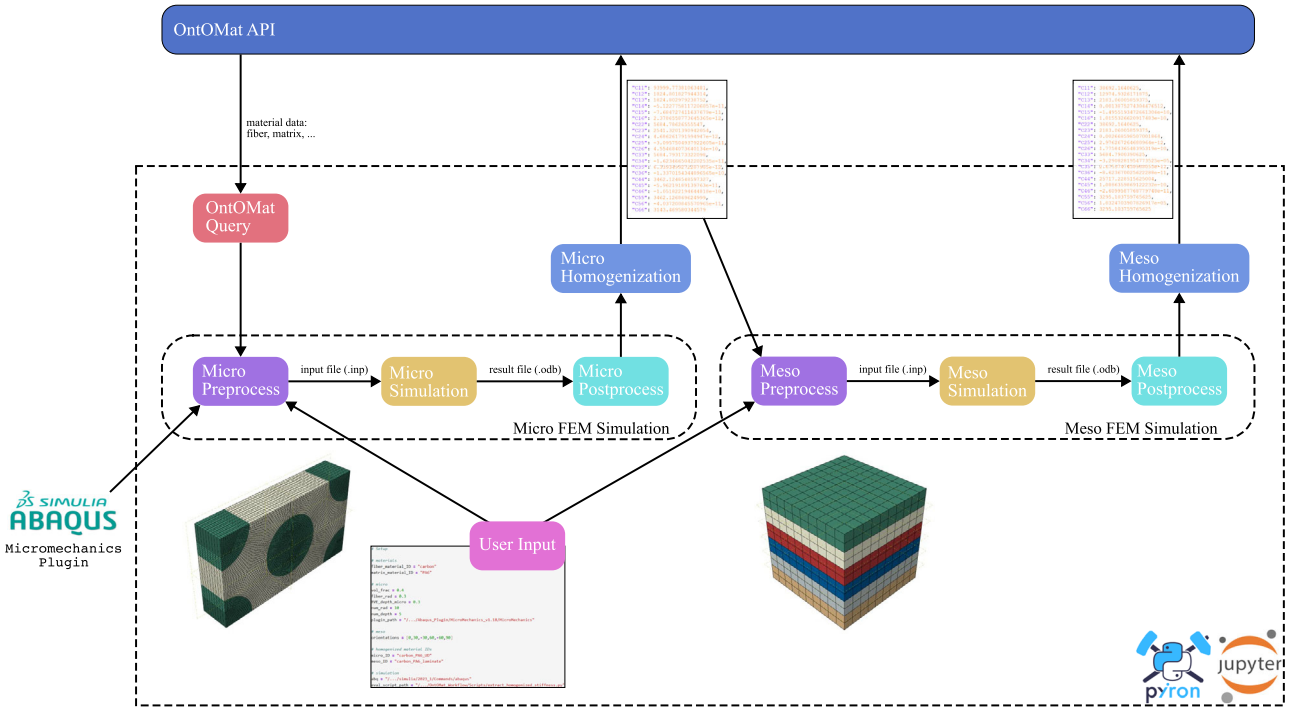


FIGURE 16 | Multiscale simulation workflow structure in pyiron.

`pyiron` nodes. These include the **FEM Solver** and the **FEM Evaluator**.

2.7.5 | Challenges and Conclusion

Several challenges were encountered during the workflow development. As mentioned above in the AnAttAl contribution, `pyiron` does not provide a native job class for `Abaqus` or direct interaction with an HPC SLURM management tool. In this case, the issue was addressed using subprocesses. Moreover, the workflow in its current version only works with a single FE solver, that is, `Abaqus`, and requires specific calls to the HPC SLURM manager to access the commercial licenses correctly. Hence, to use this workflow with a different licensing structure on another HPC, a few code lines would need to be adjusted. In a future version, it would be of great value to generalize the workflow to work with a variety of FE software tools.

The implemented workflow demonstrates how multiscale simulations can be automated with information from the OntOMat KG. In our specific case, a two-step homogenization was implemented, but further steps could be added in the same fashion. The workflow management environment `pyiron` proved to be useful to coordinate FE simulations.

2.8 | DaMaStE: 3D Microstructure Simulation Workflow for Battery Electrodes

MD Funding Phase 2

2.8.1 | Introduction

Lithium-ion batteries are essential for modern energy storage, powering applications from portable electronics to electric vehicles and grid storage. Despite their widespread use, developing and optimizing these batteries remains challenging due to the complex nature of their chemistry, which depends on numerous factors, such as the type of active material that governs electrochemical behavior, the choice of conductive additives and binders, their mixing ratios, as well as the electrode's thickness and porosity. Additionally, high production costs and time-consuming manufacturing processes add further difficulties in bringing the batteries into daily-life applications. Addressing these challenges requires advanced tools that can shorten the effort involved in experimentally designing and testing the batteries. GeoDict 3D microstructure simulation software offers a powerful tool that enables comprehensive modeling and characterization of the lithium-ion battery electrodes at a microstructural level. In the DaMaStE project, GeoDict is used to simulate the 3D structure of NCM622-based cathode electrodes (a) via segmentation of imported focused ion beam scanning electron microscopy (FIB-SEM) tomography images or (b) by virtually creating a 3D electrode structure. The simulation workflow embedded in the DaMaStE project allows detailed investigations of variations in the 3D electrode microstructure and its direct impact on electrochemical behavior. The primary objectives include analyzing porosity at multiple scales, encompassing macropores between the active material and the carbon-binder domain (CBD) as well as inner pores within the CBD, along with diffusional tortuosity.

2.8.2 | Workflow and Methods

The simulation pipeline presented (Figure 17) links a battery production ontology and database with the GeoDict simulation environment. Material and cell parameters, together with 3D microstructure data from FIB-SEM or μ CT or virtually generated structures, are used to create 3D battery microstructure models. While the workflow establishes a structured pipeline, key steps such as image processing, segmentation, and model

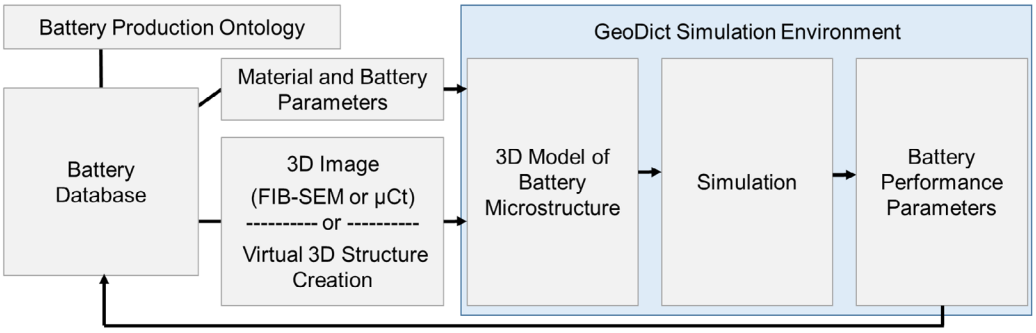


FIGURE 17 | Overview of the data-to-simulation pipeline used in the DaMaStE project for GeoDict 3D microstructure simulation to derive battery electrode performance parameters.

parameterization are not fully automated and require expert user input. The resulting models are simulated in GeoDict to derive battery performance parameters, which can be stored for comparison and analysis.

The 3D structures of the electrodes were reconstructed using advanced imaging techniques to capture the detailed microstructure (Figure 18). 3D images were obtained from X-ray computed tomography (CT) and FIB-SEM, providing high-resolution insights into the internal morphology of the materials. In addition to structural imaging, relevant material properties were incorporated into the analysis, including electrical and ionic conductivity, diffusion coefficients, and reaction kinetics, obtained either from experimental measurements or from literature sources. Properties of the electrolyte, such as conductivity, transference number, and concentration, were also considered to accurately model the electrochemical environment. To ensure reliability and accuracy, the reconstructed microstructures and associated properties were validated against experimental data, including electrical and ionic conductivity measurements, charge/discharge curves, impedance spectra, and rate performance of real cells. This combined approach allowed for a comprehensive understanding of the relationship between electrode microstructure and electrochemical performance.

In the second method, the electrode microstructure was created using the GeoDict structure generation tool. This allows much faster structure creation and the creation of structures for which the creation in the lab is challenging or impossible. Active material type (NCM) was selected as the electrochemically active phase. The solid volume fractions of the active material and the CBD were defined to reflect the targeted electrode composition. After all structural parameters were specified, the structure generation routine was executed, resulting in an automatically generated 3D electrode microstructure with spatially distributed active material particles and CBD (Figure 19). This synthetic microstructure enables systematic studies of the influence of composition and microstructural parameters on electrochemical performance.

These digital twins allow systematic variation of individual parameters, such as porous carbon additive type or binder distribution, enabling targeted studies of their influence on transport properties and electrochemical performance. The resulting artificial structures are likewise converted into voxel mesh-based 3D geometries, which are fully compatible with GeoDict's simulation modules. By combining real and artificial structure generation within a unified workflow, the implemented approach provides a robust platform for investigating microstructure–performance relationships, identifying transport limitations, and correlating simulation results with experimental observations.

Reconstructed (FIB-SEM/µ CT-based) and virtually generated electrode microstructures were converted into voxel-based 3D geometries in GeoDict and assigned to the phases active material, CBD, and pore/electrolyte. Transport simulations were performed on the 3D domains using consistent phase-specific material parameters and boundary conditions across all cases. The simulations yielded structure- and transport descriptors such as phase volume fractions/porosity, pore-related metrics, effective transport properties, and diffusional tortuosity along the principal directions.

Simulation results were evaluated by extracting key microstructural and transport descriptors from the 3D electrode domains. Reported metrics include phase volume fractions and porosity, pore-size-related measures, and connectivity or percolation of the solid phases. Transport properties were quantified via

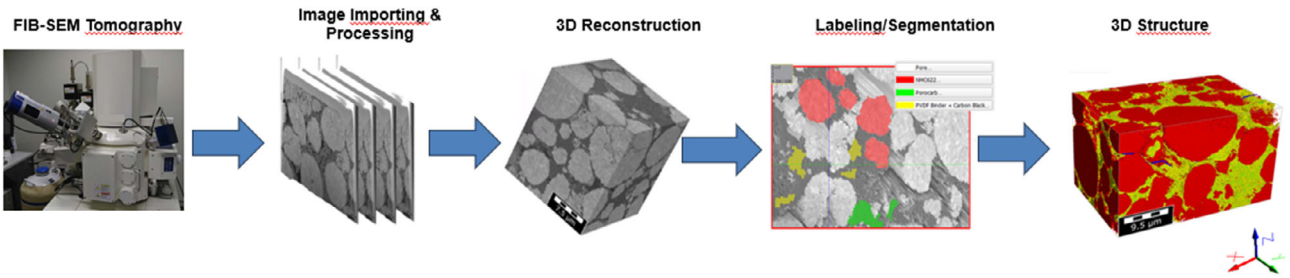


FIGURE 18 | Image-based 3D microstructure reconstruction in GeoDict from FIB-SEM tomography, including import and preprocessing, 3D reconstruction, phase segmentation, and generation of a voxel-based electrode structure.

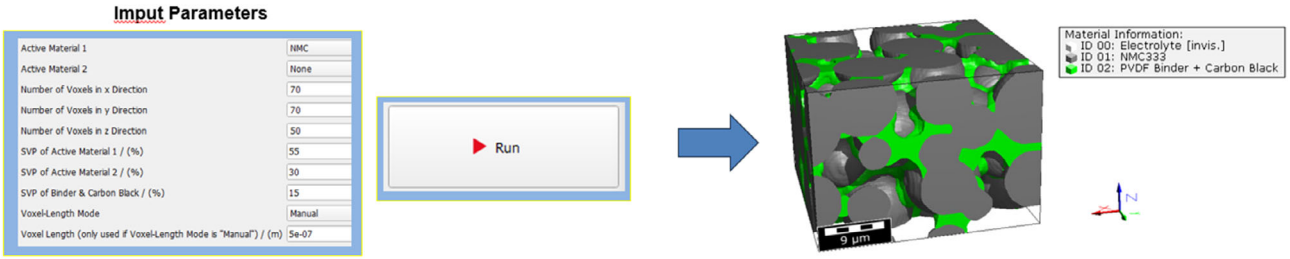


FIGURE 19 | Virtual 3D electrode microstructure generation in GeoDict, showing parameter definition and the resulting synthetic electrode structure.

diffusional tortuosity, effective ionic conductivity, and effective electronic conductivity. In addition, electrochemical battery simulations were performed at multiple C-rates, and the resulting rate performance was assessed using the simulated charge–discharge response and capacity retention as a function of rate. Results can be stored together with the corresponding input parameters in the project database to enable consistent comparison across samples and simulation scenarios.

2.8.3 | Summary, Challenges, and Outlook

This work presents a GeoDict-based 3D microstructure simulation workflow for NCM622 cathodes that combines image-based reconstruction and virtual structure generation. The approach enables systematic studies of how microstructural features influence transport properties and electrochemical rate performance and supports structured comparison across different microstructures and parameter sets.

Current challenges are mainly related to the limited degree of automation and the uncertainty associated with microstructure reconstruction and model parameterization. Image processing and segmentation remain sensitive to user choices and data quality, and some phase properties and boundary conditions must be defined based on assumptions or external measurements. In addition, computational cost increases strongly with domain size and resolution, which constrains the achievable parameter space.

Future work will focus on increasing reproducibility through standardized preprocessing protocols, improved segmentation and parameter identification, and tighter coupling of simulation outputs with experimental validation. Extending the workflow toward more automated, data-driven microstructure generation and calibration is expected to further accelerate the exploration of microstructure–performance relationships.

2.9 | DigiChrom: Simulation-Based Characterization Workflow for Electroplated Coatings

MD Funding Phase 2

2.9.1 | Introduction

In technical applications, the optimization of electroplated coating systems is commonly pursued through experimental trial-and-error approaches. However, electroplated coatings exhibit complex process–structure–property relationships whose systematic experimental investigation is highly demanding and often provides only limited mechanistic insight. Numerical simulations, by contrast, enable access to material information that is experimentally inaccessible or obtainable only with considerable effort, such as local stress and strain distributions in heterogeneous microstructures or interfacial stresses that critically influence coating performance. Moreover, owing to the intricate interdependencies between process parameters, microstructure, and material properties, data-centric approaches that enable the application of ML methods are particularly promising for identifying governing mechanisms and correlations. This creates a strong demand for digital frameworks that integrate experimental and simulation data within a semantic graph database, thereby enabling knowledge-driven design of electroplated coating systems and the application of AI methods [45].

2.9.2 | Use Case and Approach

Using Cr(III)-based coatings as a representative use case, the DigiChrom project aims to facilitate a more targeted and systematic investigation of electroplated systems by formalizing domain knowledge through ontologies and by establishing traceable workflows. These workflows are designed to consistently capture experimental and simulation data, enrich them with comprehensive metadata, and make them accessible via a graph database (Figure 20). The graph database semantically organizes all experimental and simulation data according to a domain ontology developed specifically for hard chromium coatings in the DigiChrom project and implemented in accordance with the FAIR principles [46]. This ontology-based data model supports interoperability and enables experimental and simulation data to be consistently linked to corresponding samples and process parameters, thereby supporting their subsequent use in ML applications.

The enrichment of the materials data space for the optimization of electroplated coating systems with simulation data requires fully traceable and reproducible workflows. This is particularly important because simulations depend on experimental inputs such as microstructural statistics and experimental data obtained from coatings produced under well-defined process conditions, with results from individual simulation steps serving as inputs for subsequent stages. To address this need, a workflow for simulation-based coating characterization (indicated by the gray box in Figure 20) is being established. This workflow uses the graph database both as a source for experimental data and as a sink for simulation models and results. The consistent representation and

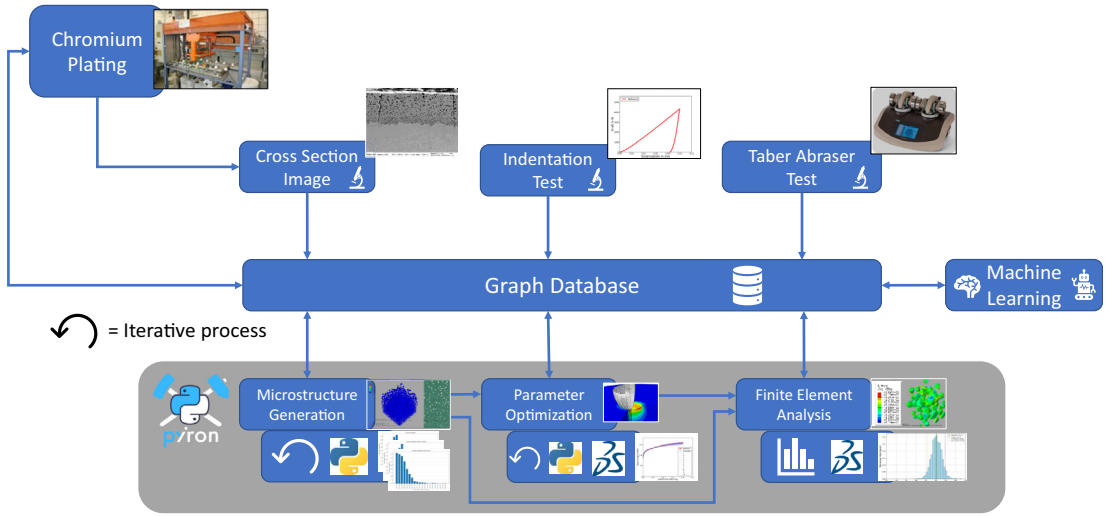


FIGURE 20 | Use case developed within the DigiChrom project for the knowledge-driven optimization of electroplated chromium coatings, highlighting the workflow for simulation-based characterization shown in the gray box.

structuring of data support AI readiness by making the data easier to use in subsequent AI applications. They may also facilitate future automated orchestration, although an agent-facing interface is not part of the workflow described here.

2.9.3 | Implemented Workflow

The simulation workflow is implemented using the workflow management framework pyiron, as it enables the integration and connection of different simulation tools within a unified workflow environment and is additionally supported by the PMD. For each individual chromium coating instance, pyiron orchestrates the automated retrieval of experimental data from the graph database, executes the simulation steps, and supports reproducible workflow execution. The experimental data comprise cross-sectional microstructural imaging, (instrumented) hardness measurements, and wear testing using a Taber Abraser (Figure 20). Starting from 2D cross-sectional microstructural images and derived statistical descriptors such as those characterizing microcracks or dispersed hard particles in dispersion coatings, statistically equivalent 3D microstructures are generated. Dedicated Python-based software tools have been developed to create the corresponding geometries, which are subsequently meshed and used to construct FE models of the coatings. Based on these microstructure-resolved models, the mechanical behavior of the coatings is identified through inverse FE modeling driven by indentation experiments. In this parameter identification workflow, elastoplastic material model parameters are calibrated using instrumented hardness measurements with spherical indenters [47, 48], while fracture-related properties, such as fracture toughness, are to be inferred from hardness measurements using sharp indenters [49]. The parameter optimization is performed using Python-based tools that link simulated and experimental indentation responses and minimize a least-squares-type objective function. FE simulations were conducted using the commercial FE software ABAQUS (Dassault Systémes). The results of microstructure generation and material parameter identification are subsequently combined in FEA under defined loading scenarios. Representative use cases include simulations of tribological loading conditions, such as those encountered in Taber Abraser tests. These simulations provide detailed insight into local stress and strain distributions within heterogeneous microstructures, including the effects of dispersed particles and microcracks, which are difficult or impossible to resolve experimentally.

2.9.4 | Summary and Outlook

Overall, the presented workflow covers the entire simulation-based characterization of electroplated coatings, ranging from microstructure-based model generation and material parameter identification to experimentally informed FEA analyses, with a semantic graph database serving as the central data backbone. At the time of submission, the workflows for simulation-based characterization of chromium coatings were available in a documented and fully specified form.

Reference implementations (e.g., microstructure generation, parameter identification, and automated postprocessing) are under continuous development and are openly available in the GitHub repository, https://github.com/CMD-HSO/digichrom-sim-workflow.

2.10 | DiMoGraph: Active Learning Workflow for Graphene-Based Macromaterials

MD Funding Phase 2; Updated Contribution

DiMoGraph studies graphene-based macromaterials, aiming to describe the relation between the structure (graphene flake properties and packing density) and the electrical conductivity of the macromaterial [50]. Such macromaterials are simulated using a network model that replaces the real system with a network of conductors representing either the transport of a graphene flake or the transport between two flakes. Because graphene-based macromaterials contain many flakes, the resulting network and its computational cost become cumbersome. Surrogate models provide an alternative, purely mathematical, description of the material. To incorporate the underlying physics, they were trained using results from the network model. In addition to

developing models, DiMoGraph aims to make its models and data ready for AI-driven materials science. To this end, the project has published a training workflow that employs a workflow manager and ontology-based data storage, which can be found in the PMD workflow store [51]. The network model in the uploaded workflow is a simplified version of the network model previously used [52–55]. The workflow is shown schematically in Figure 21; its main features are discussed below.

2.10.1 | *Workflow Manager*

The workflow is implemented using the workflow manager pyiron [3] to facilitate the training in a standardized and well-documented manner. It uses the class-based version of pyiron. Two jobs have been implemented using pyiron, which are each based on a pyiron class: `NetworkModelJob` and `AdaptiveGaussianProcessRegressionJob`, which interact, as illustrated in Figure 21. `NetworkModelJob` is used to create training data, while `AdaptiveGaussianProcessRegressionJob` performs ML. Both jobs interlink with an ontology, as described below.

2.10.2 | *Creation of Training Data*

`NetworkModelJob` runs the network model, which is illustrated in Figure 21. Based on input parameters (the conductivities of the connections in the network and its topology), the job calculates the total conductivity using nodal analysis. `NetworkModelJob` can be used independently of the discussed workflow. In this case, the user can control the workflow using a Jupyter Notebook. Many of the implementation details are hidden inside the pyiron classes. In addition, input and output data are standardized and also stored in an ontology automatically as discussed below.

2.10.3 | *Training Process*

The training of the surrogate model utilizes an active training approach, which was already described in [4]. To achieve this, a calculation with the above-mentioned network model is performed for a point of a predefined parameter space, and the result is added to the training dataset. Then, a Gaussian process regression model is trained, which is carried out by the `GaussianProcessRegressor` of the `scipy` library. A prediction of the model for the entire parameter space is calculated together with an uncertainty prediction. For the point with highest uncertainty, another network model calculation is performed. Figure 21b shows how iterative training reduces the deviation by adding more and more training data. In Figure 21c, sample functions, which are used within the Gaussian process regression, are plotted together with the training data. They strongly deviate in the beginning of the training but soon approach the correct values.

2.10.4 | *Storage to an Ontology*

Simulation results are stored using a domain ontology developed as part of the DiMoGraph project [50, 56, 57]. The domain ontology is based on PMDco version 2. It stores not only the data from the simulation but also the various steps from the training process. The interaction with the ontology is realized using `owlready2`. In case of `NetworkModelJob`, a dedicated method `to_ontology` is called after the network simulation to write

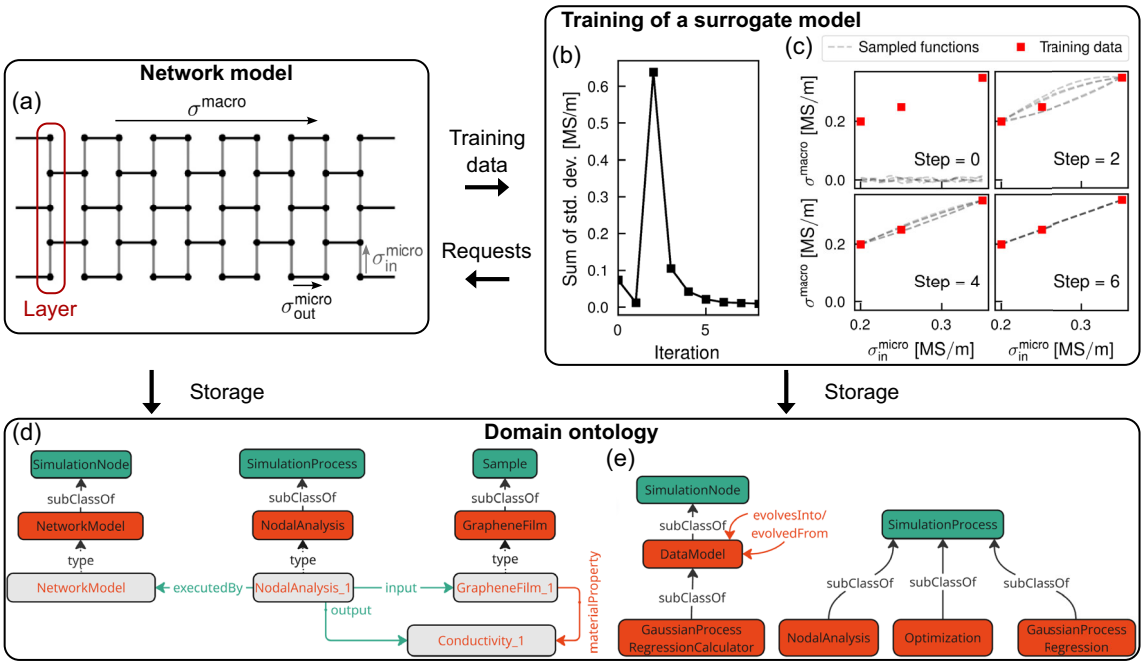


FIGURE 21 | Models used in the DiMoGraph project. (a) A network model is used to calculate the macroscopic conductivity σ^{macro} from structural information (the topology of the network) and microscopic conductivities ($\sigma_{\text{in}}^{\text{macro}}$ and $\sigma_{\text{out}}^{\text{macro}}$, describing transport inside and in-between layers, respectively). (b) The error reduction during the training of the surrogate model is shown. The training relies on the network model to provide training data. (c) Sampled functions of the Gaussian process regression, which approach the training data. Only the part of the parameter space for $\sigma_{\text{out}}^{\text{micro}} = 100\,\text{MS/m}$ is shown. (d and e) The data from the network model as well as the training process are stored into a specialized domain ontology, from which small snippets are depicted.

the results into the ontology. In case of `AdaptiveGaussianProcessRegressionJob`, the interaction with the ontology is intertwined with the training and defined in the method `run_static`.

2.10.5 | *User Interaction*

When running the workflow, the user only interacts with `AdaptiveGaussianProcessRegressionJob` by defining the input parameters of the workflow. These input parameters contain the relevant features for the training, the parameter space of interest, and the numerical parameters of the training algorithm. The name of the pyiron job used to create training data can be specified, too, which could be used in principle to adapt our workflow to other suitable pyiron jobs. The entire training and the interaction with the ontology are hidden from the user. A possible visualization of the training process can be found in the uploaded workflow [51].

2.10.6 | *Summary*

The project developed an automated pyiron workflow for network-model simulations and surrogate-model training. This work contributes to digitalized materials science by facilitating the integration of different tools. Coupling the workflow with ontological data storage enables semantic search and further improves the accessibility and usability of both the workflow and its generated data for AI applications. The structured data could also support future agentic systems, although no agent-facing interface is implemented in the workflow described here.

2.11 | InSuKa: Intelligent Search Engine for Finding Optimized Rubber Compounds

MD Funding Phase 2

Rubber compounds are produced in a discontinuously working internal mixer with the addition of components such as rubber, fillers, plasticizers, and chemicals. To ensure high quality, mixing instructions must be developed for each recipe. Optimal process settings are currently determined iteratively based on expert knowledge. As the scale-up from lab to production machines is not possible today, these process developments take place on production machines, which are limited in terms of time, costs, and material.

To address this challenge, as part of the InSuKa project, a search engine is being developed in which the user selects the product with the corresponding quality requirements and the available equipment (e.g., mixer and roller mill) (Figure 22). The results returned are the recipe to be produced and the associated mixing instructions. The basis for developing the mixing instructions is provided by laboratory mixing trials, which are used to predict the mixing curves.

The workflow for developing the mixing instructions can be taken from Figure 23.

First, mathematical models are developed on the basis of laboratory mixing trials, which represent the development of the mixing curves and the mixing quality over the mixing time. The mixing curves describe the time-dependent development of process variables, such as the ram path and the torque and are recorded during each mixing process. The mixing quality is characterized by the amount of incorporated and dispersed carbon black and the distribution of the mixture components.

To model the mixing process, superior functions are defined for the mixing curves (ram path, torque, and batch temperature) and for the mixing quality (incorporation, dispersion, and distribution). The parameters of these functions can be adjusted based on the recipe used (polymer type, filler type, amount of filler, etc.) and the process parameters (rotor speed, rotor and mixing chamber temperature control, and fill factor). The developed models can be used to identify optimized recipe and process parameters, resulting in a reduction in mixing time and energy input. In addition, further influencing factors will be integrated into the model to transfer the results obtained on a laboratory scale to the production mixer. This extension is expected to reduce mixing time, energy input, and waste by reducing the need to develop mixing instructions for the production mixer.

The algorithms developed to predict and optimize mixing curves are integrated into the InSuKa app, providing a tool to develop optimized mixing instructions based on the selected recipe.

2.12 | DIPONI: AI-Based Optimization of Control and Decision-Making for Rubber Extrusion

MD Funding Phase 3; Project Started in 2025

Building on the preceding project DIGIT RUBBER, DIPONI aims to develop an optimized and resource-efficient rubber extrusion and continuous vulcanization process using AI-based process control and quality prediction. Furthermore, to optimize not only the process itself but also the efficient usage of material, an additional aspect of the AI consists of assisting users with

FIGURE 22 | Development of a search engine to find optimized rubber compounds.

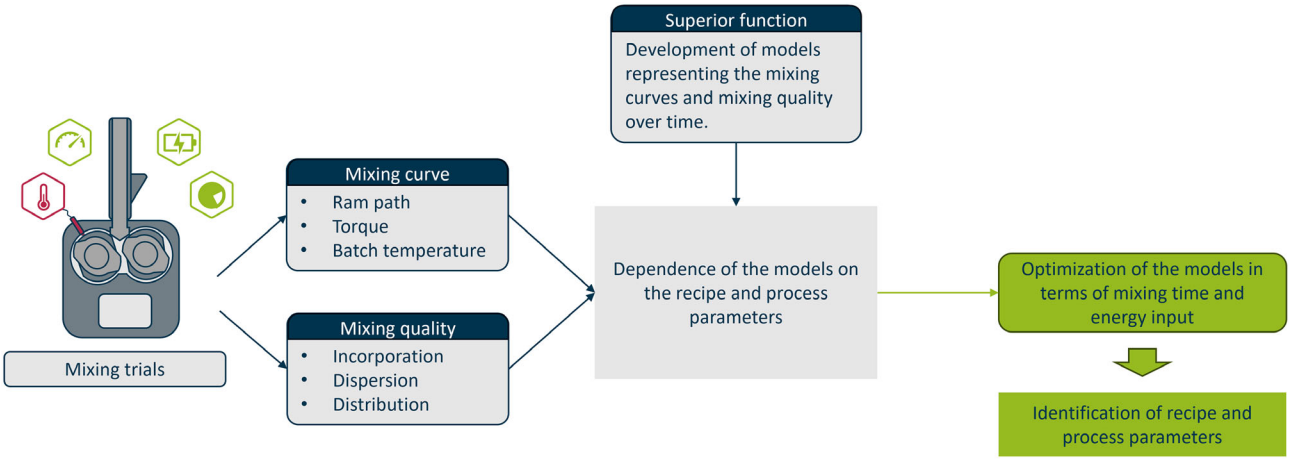


FIGURE 23 | Development and optimization of models for the rubber mixing process.

business-related decisions. The developed algorithms and workflows should also be transferable to similar processes and other material classes, such as polyacrylate processing in adhesive production.

2.12.1 | *Technical and Business AI*

For the technical part of the AI, models will be trained on real process data to predict quality deviations along the process chain. A distinction is made between defects detected before and after vulcanization. Unvulcanized semifinished parts with deviations in geometry or surface quality can be returned to previous process steps, improving material efficiency and reducing waste. Vulcanized parts that do not meet the quality requirements must be recycled through additional processing steps before the material can be reused. Continuous AI-based monitoring of the relevant process parameters is planned to complement this quality prediction. If unfavorable process conditions or faulty products are predicted, the planned AI controller would adjust machine parameters in real time with the aim of reducing faulty output and unnecessary material and energy consumption. As this part of the AI is focused on controlling the system in real time, the long short-term memory (LSTM) architecture is considered a promising option because it can take past system states into account in its predictions. This assessment is supported by reported applications of LSTMs in polymer science [58]. The digitalized extrusion line with its corresponding data flows is shown schematically in Figure 24.

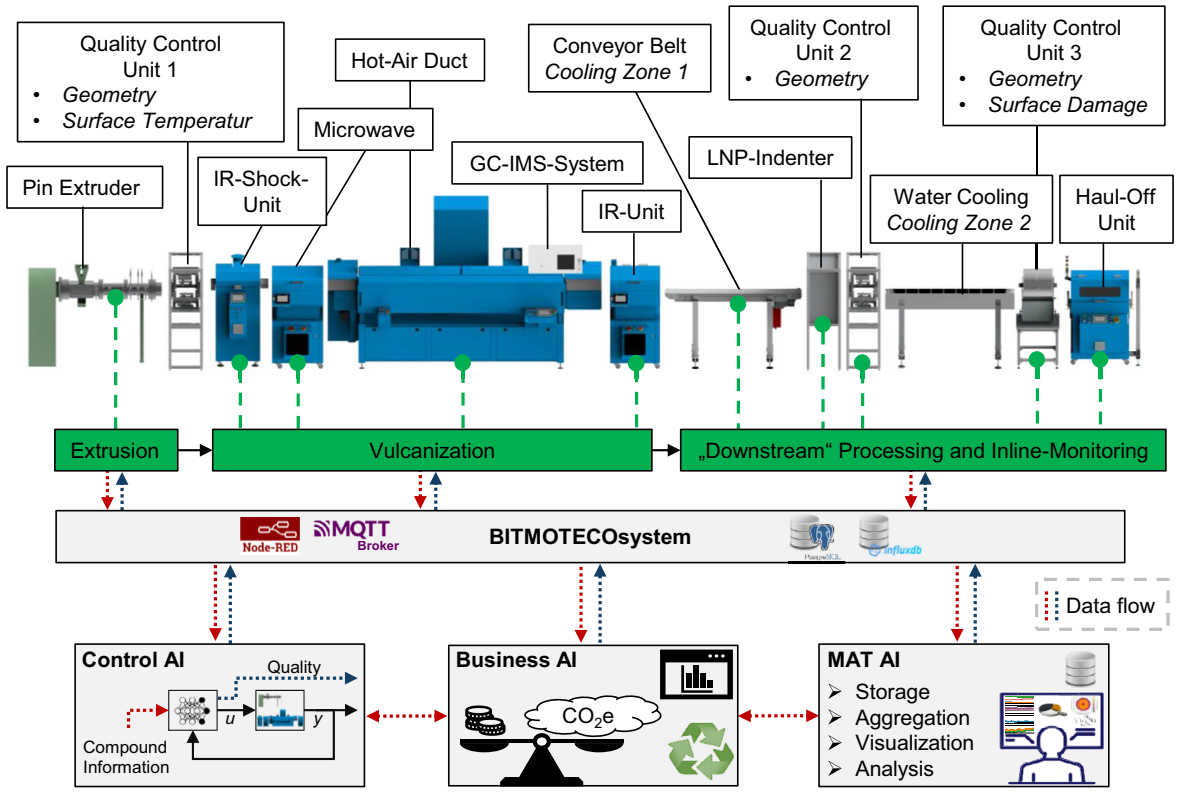


FIGURE 24 | Final extrusion line and data flows.

The planned technical AI systems are to be extended with production-related, environmental, and economic data from production planning, enterprise resource planning and environmental, social, and governance management. This additional meta-level, hereafter referred to as the "business AI," considers not only technical quality criteria but also economic and environmental production objectives. The combined system is intended to support optimization of material recovery, energy consumption, inventory levels, and quality requirements, including trade-offs between these objectives. In Figure 25, the interaction between technical AI and business AI with the different aspects of the manufacturing process is illustrated.

2.12.2 | Workflow and Demonstrator

Following the initial concept development, the existing system, process, and data landscapes of the participating organizations were systematically assessed. Structured workshops with the project partners were used to document the current processes, IT systems, production equipment, and associated data inputs and outputs. The objective was to identify similarities, differences, weaknesses, and requirements for the demonstrator.

The analysis showed comparatively similar process structures between the institute and manufacturing participants for the class of rubber, whereas the one using another material class (polyacrylate) differs significantly. Considerable differences were also identified regarding data maturity and availability. The research institute does not currently operate a system landscape directly comparable to that of a typical industrial company. Relevant elements of an industrial system landscape must therefore be simulated within the demonstrator at the institute to support evaluation of the transferability of the developed concepts, data models, and AI solutions.

The demonstrator focuses on the data-driven optimization of manufacturing indicators such as inventory levels, delivery reliability, throughput time, capacity utilization, quality, and productivity. Environmental and sustainability-related indicators are additionally considered to support the ecological optimization

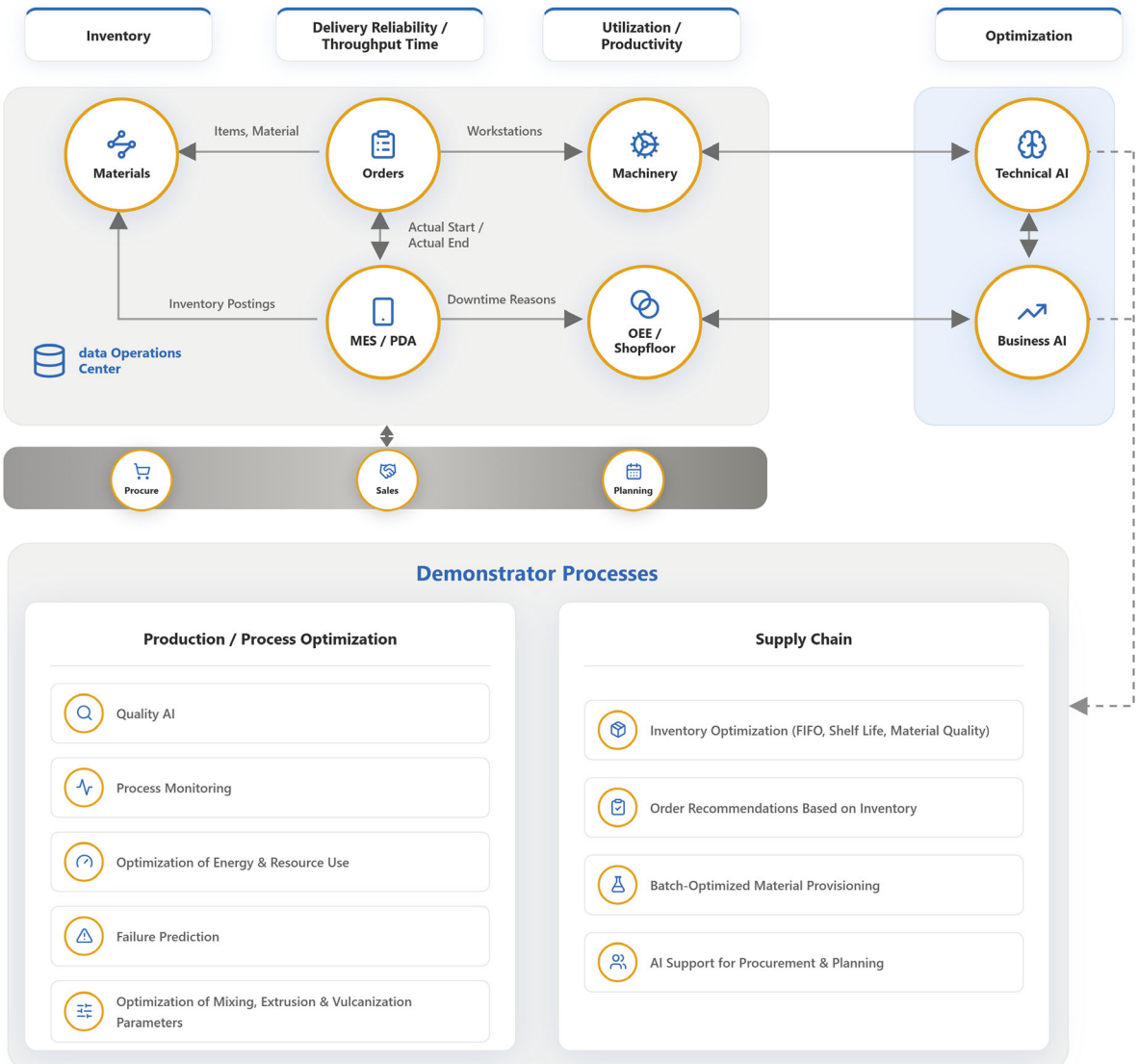


FIGURE 25 | Interconnectivity within the manufacturing process.

of products and production processes. By combining technical, economic, and ecological data, the demonstrator addresses different levels of industrial decision-making within one integrated approach.

In supply chain management and enterprise resource planning, possible applications include inventory optimization considering first-in-first-out principles, shelf life, and material quality. Further planned functionalities comprise the generation of order proposals, batch-optimized material provision, and AI-based support for purchasing and production planning. The business AI is intended to support lower inventory levels while meeting delivery and quality requirements and improving the utilization of available material batches.

Several AI functionalities are being evaluated for integration into the process and data model. Initial tests indicate that DecisionTree and XGBoost may be suitable approaches for the discussed business AI optimization problems. Other candidates include deep-learning methods such as feed-forward neural networks (FNNs) and reinforcement-learning (RL) approaches. These candidates will be tested and evaluated with data from the demonstrator. Potential production and process applications include automated quality analysis, continuous process monitoring, the prediction of disruptions and equipment failures, optimization of energy and resource consumption and the operational parameters of extrusion and vulcanization.

2.12.3 | Interoperability Challenges

A further objective is the development of a transferable, centralized, and interoperable data model. Due to the heterogeneous process structures, data environments, and system landscapes of the partners, a common data foundation is required to harmonize different data sources and support the development of transferable AI models. Existing systems, machines, and data sources should be connected through standardized interfaces without extensive application-specific software changes. The initial architecture has already been conceptually defined and is currently being evaluated and refined through clickable prototypes in parallel with the demonstrator development. The demonstrator is therefore intended as a flexible and scalable framework for heterogeneous industrial environments rather than as a company-specific solution.

The transfer of the combined AI systems to different machines, processes, and material classes creates two major interoperability challenges: the general applicability of the AI models and the development of consistent workflows. The project is addressing both challenges through its emerging data and workflow architecture. AI models can only be applied reliably within application ranges for which they have been trained and validated unless additional inline training is performed. Therefore, two scenarios are considered. Either a generalized AI model is trained in advance for different processes, or an automated training pipeline creates an application-specific model before deployment.

The first approach requires a unified workflow, a common data foundation, and an ontology translating individual processes into a standardized structure. In accordance with the FAIR principles, common workflows, data models, and ontologies improve the exchange and comparability of data and results. However, harmonizing processes, terminology, machine configurations, data quality, and operational objectives is challenging even within the same material class and becomes more difficult across different classes of material and specifications.

The second approach represents a form of automated ML (AutoML; see for example, [59]) and is comparatively easier to establish. Application-specific models may perform better because they account more precisely for individual process conditions and data structures. However, without a unified workflow, comparability and data exchange are reduced. DIPONI therefore aims to combine standardized and interoperable workflows with the possibility of adapting and training AI models for specific applications. Both approaches are expected to support improvements in material efficiency, product quality, process stability, and energy consumption.

At the end of the project, the developed control and classification algorithms will be implemented in pyiron, a Python-based workflow environment for materials science. This is intended to transform individual AI scripts into reproducible workflows in which process, training, and evaluation steps can be documented, reused, and transferred between applications. The resulting algorithms and workflows will be made accessible through the PMD workflow store.

2.13 | GlasAgent: LLM-Driven Agent Interface for Atomistic Glass Simulation

MD Funding Phase 3; Project Started in 2025

The GlasAgent project aims to accelerate specialty glass development by introducing process automation together with an agentic material development framework across the glass development loop from simulations through robotic melting to automated sample characterization. Through a conversational interface powered by a large language model (LLM), glass developers can formulate design objectives, prompting the agent to select and execute the appropriate tools (Figure 26a). In the simulation domain, expert knowledge is encapsulated in specialized tools providing direct access to data-driven modeling, thermodynamic calculations, and atomistic simulation of specialty glasses (Figure 26b). The atomistic glass simulation workflows (Figure 26c) are the topic of this contribution.

2.13.1 | Motivation and Goals

The atomistic simulation of glasses is challenging: glasses are amorphous, requiring large simulation cells and out of thermal equilibrium, that is, their properties depend not only on the composition but also on the thermal history of the sample. Atomistic simulations therefore face major challenges in industrial glass development. Historically, accurate ab initio methods were too computationally expensive and classical force fields were often not flexible enough to capture the multitude of interactions in industrially relevant multicomponent glasses over a wide range of temperatures. Today, new, flexible machine-learned interatomic potentials with millions instead of tens of parameters

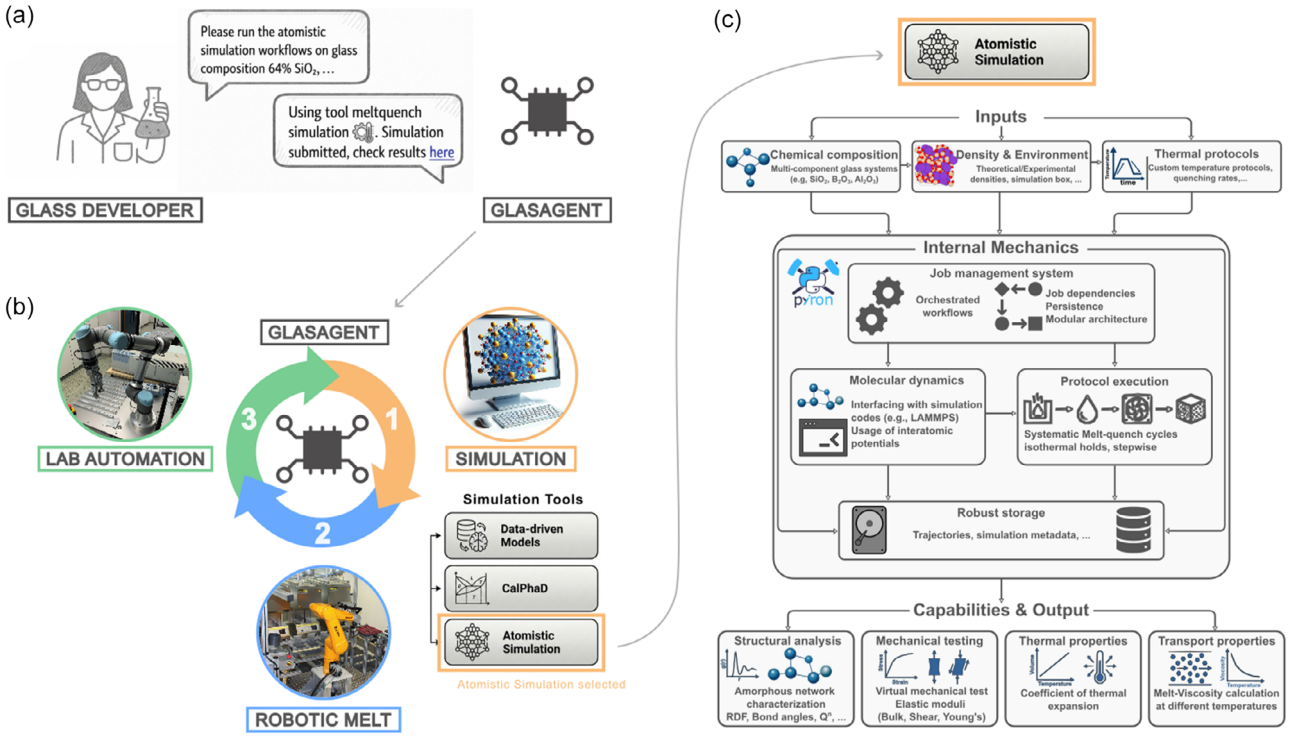


FIGURE 26 | (a,b) A glass developer formulates a request through a chat interface, prompting the GlasAgent orchestrator to select and execute the appropriate tool(s) from the specialty glass development loop. (c) Atomistic simulation workflow from simulated melt-quench through structure analysis and property calculations.

and—in principle—coverage of the entire periodic table promise to close this gap for the first time.

Besides the challenge of a favorable tradeoff between model accuracy and computational cost, there is also a challenge of time-to-result: atomistic simulations of glasses typically require expert knowledge, use-case-dependent simulation parameters, and sometimes access to HPC. From the glass developer perspective, this translates to a time to result of the order of weeks, which is incompatible with the development cycle of industrial glass.

The GlasAgent project aims to tackle these challenges as follows:

1 By making curated atomistic simulation workflows available to experimental glass developers via a natural language interface, we aim to bring time-to-result down to "overnight," enabling the integration of atomistic simulations into the daily glass development routine for the first time.

2 By moving from classical potentials to universal machine-learning interatomic potentials, we aim to improve the *accuracy* of the atomistic simulation toolbox.

2.13.2 | *Workflow Design and Implementation*

The GlasAgent project introduces workflows that cover all stages required to go from a glass composition to the properties of the resulting glass. As illustrated in Figure 26c, typical workflow steps include the generation of initial configurations, high-temperature melting, controlled quenching, equilibration, and subsequent property evaluation at relevant temperatures. The automated analysis routines extract structural descriptors at both short and medium-range scales, such as radial distribution functions, bond angle distributions, and ring-size distributions. Simulations of macroscopic properties, such as elastic moduli, thermal expansion coefficients, and viscosity estimates, are implemented.

The execution of the simulation workflows is managed by the pyiron workflow framework [3, 60], which enables the seamless coupling of external molecular dynamics engines, such as LAMMPS [61], with Python-based pre- and postprocessing tools. The choice of pyiron as the workflow framework is motivated by its modular architecture and Python foundation, which simplifies the integration of new simulation tools, data analytics methods, and machine-learning frameworks.

The workflows have been released under an open-source license in the amorphousPy [62] Python package.

2.13.3 | *Agentic Use of Workflows*

In addition to the workflows mentioned above, the amorphousPy [63] Python package ships a web server written in FastAPI. This web server exposes endpoints for submitting and managing simulation jobs as well as for retrieving and visualizing results.

The documented endpoints are available in the form of a REST API as well as in the form of "tools" according to the Model Context Protocol (MCP) specification. This allows any compatible LLM frontend to connect to the MCP server and start launching simulations. When connected to a chatbot user interface, such as ChatGPT, a typical conversation would include:

submitting a job, polling its status, retrieving a summary of results, and surfacing the URL to the accompanying interactive visualization of the atomic structure, radial distribution function, viscosity curve, etc.

On the execution side, each request is expanded into a dependency-aware pipeline, for example, structure generation, LAMMPS melt-quench, and the requested property analyses, which are orchestrated by executorlib from the pyiron ecosystem. Executorlib's pluggable executors let the identical pipeline definition run as local subprocesses during development or be dispatched to a SLURM/Flux cluster for production workloads, with no change to the workflow code. Submissions, settings, and results are persisted in a job store keyed by a hash of the composition and simulation parameters, so identical requests are served from cache, jobs remain queryable and resumable across sessions, and all metadata stays available for downstream machine-learning use.

2.13.4 | Challenges and Outlook

The atomistic simulation of real-world glasses in the computer always involves a series of simplifications, such as the finite cell size, high cooling rate, approximate interatomic potential, etc. Judging which approximations are admissible for a given simulation goal depends on a great deal of context and has traditionally been the task of a simulation expert. Can this expertise be cast into a workflow operated by a LLM? Although this may not be feasible in general, well-scoped tasks can benefit from substantial gains in speed and scalability compared with those of traditional approaches. The project will therefore continue to pursue this direction.

Much work remains in the project. The current workflows rely primarily on empirical interatomic potentials, which cannot capture the full chemical complexity of advanced multicomponent glasses. Extending the framework to machine-learned interatomic potentials, particularly universal models trained across the periodic table, is therefore a key next step. Such potentials offer the prospect of combining high accuracy with computational efficiency, enabling realistic simulations of large systems where both short- and medium-range order are critical.

Another major challenge lies in usability. To be effective in industrial contexts, workflows must be operable by users without prior experience in atomistic simulations. This requires automated determination of simulation parameters, convergence checks, and quality control procedures that eliminate reliance on expert "tuning." Input parameters that are opaque to nonexperts must be replaced by adaptive algorithms informed by prior simulations and domain knowledge.

Future developments may incorporate learning mechanisms that allow the agent to improve continuously. By leveraging accumulated simulation and experimental data, the system could refine models, optimize workflow parameters, and enhance predictive accuracy over time. Ultimately, this capability could enable on-the-fly learning within the digital twin, adding a data-driven, self-improving approach to further accelerate the glass development process.

3 | Conclusions

The article distinguishes the workflow-management layer from the application-oriented workflows built for individual scientific and industrial contexts. Within PMD, pyiron and SimStack are the canonically supported workflow environments. The thirteen project contributions demonstrate four complementary roles of digital workflows in materials research and engineering. The projects dealing with data acquisition and FAIR data storage transform heterogeneous experimental, simulation, and production data into structured and reusable resources. The simulation-automation contributions encapsulate software execution, parameter studies, model calibration, and access to computational infrastructure. The multiscale contributions transfer information between models or representations associated with different structural, physical, temporal, or process scales. The AI/ML-driven contributions range from directly integrated learning, optimization, and agent-mediated operation to applications in which such capabilities remain partial or planned. These groups describe the principal focus of each contribution rather than mutually exclusive categories.

Each contribution addresses a different scientific scope and therefore emphasizes a different combination of data structures, artifact management, orchestration, scale transfer, learning, and automated interaction. AI-readiness is used here to make these complementary contributions visible: a workflow may support AI-readiness by directly integrating learning or optimization but also by generating traceable data, automating reliable simulations, or preserving the context required to transfer results between models.

3.1 | Complementary Pathways to AI-Readiness

The projects in the data-acquisition and FAIR-data-storage group show that AI-readiness begins before model training. DigiMatUS transforms heterogeneous experimental and process data into standardized and ontology-aligned representations, stores them in structured resources such as HDF5, and connects them to neuro-symbolic analysis. HybridDigital separates parsing, semantic mapping, persistent storage, provenance, analysis, and API-based prediction to create reusable data products and a digital part record. DiStEL implements mapping steps for a semantic data-acquisition pipeline and is developing an event-driven pipeline and a downstream multiscale simulation chain. Together, these contributions demonstrate how explicit data and metadata specifications, persistent storage, provenance, and reusable artifacts make experimental and process data accessible to subsequent automated analysis.

The simulation-automation group demonstrates how workflow systems turn heterogeneous software components into explicit and repeatable procedures. DigitalModelling links experimental data, constitutive models, optimization, simulation, validation, knowledge-graph storage, and machine-readable material cards. DiMad combines deterministic input generation, legacy-code integration, parallel parameter studies, and automated validation in reusable pyiron and SimStack workflows. AnAttAl couples Abaqus, an external Fortran solver, experimental inputs, and result management in pyiron, thereby supporting reproducible

parameter studies and the generation of traceable data for future surrogate models. Together, these contributions demonstrate how workflow systems encapsulate heterogeneous software execution, parameter studies, model calibration, and access to computational infrastructure.

The multiscale-integration group highlights that automated execution alone is insufficient when data pass between models, representations, or scales. OntOMat queries constituent data from a KG, performs finite-element homogenization at successive scales, and writes effective properties and provenance back into the semantic infrastructure. DaMaStE transforms image-based or virtually generated cathode microstructures into transport simulations, although important reconstruction and validation steps still require expert intervention. DigiChrom connects experimental microstructure data, statistically equivalent 3D reconstructions, finite-element simulations, inverse parameter identification, and tribological loading cases. These contributions show that AI-ready scale bridging depends on executable coupling and explicit control of units, assumptions, boundary conditions, and provenance. Independent transfer validation, automated consistency checks, and uncertainty propagation nevertheless remain unevenly implemented.

The AI/ML-driven optimization group illustrates different levels and forms of interaction between automated decision methods and workflow execution. DiMoGraph integrates simulation-data generation, Gaussian-process regression, uncertainty-guided active learning, and ontology-based storage in a closed iterative workflow. InSuKa uses mathematical process models to predict and optimize rubber recipes, process parameters, mixing time, and energy use through an application-oriented interface, although adaptive or programmatic workflow invocation is not documented. DIPONI addresses quality prediction, process optimization, and AI-based control for polymer processing, but several data, workflow, and control capabilities remain objectives or prototypes rather than completed implementations. GlasAgent exposes curated atomistic glass workflows through documented REST and MCP interfaces and an LLM-based orchestration layer, while machine-learned interatomic potentials, adaptive parameter selection, and self-improving operation remain future work. The group therefore spans implemented optimization and active learning, application-level decision support, agent-mediated workflow invocation, and planned AI-controlled operation.

3.2 | Shared Requirements and Interoperability

Several aspects recur across the four groups and confirm that the categories are not strict boundaries. DiStEL and DigiChrom connect data acquisition with scale bridging, DigitalModelling combines simulation automation with optimization and reusable artifacts, and DiMoGraph couples active learning with structured data and provenance. The resulting profiles are therefore better understood as descriptions of emphasis rather than as labels that restrict a project to one category.

A further common observation is that stable data structures, persistent artifacts, executable orchestration, and scientific validation are broadly useful across almost all workflow types. Adaptive operation and agent-accessible interfaces are more specialized and are intentionally concentrated in contributions whose scope includes feedback, automated control, or agent interaction. Where broader machine access is useful, documented and versioned APIs, explicit input and output schemas, predictable error handling, and suitable authorization mechanisms can make existing workflow services easier to integrate into platform architectures or to expose through agent-oriented protocols such as MCP without redesigning the scientific implementation.

The results also qualify the idea of workflows as a shared abstraction layer. The abstractions are currently strongest within individual projects or technical ecosystems. Workflow environments can hide implementation details and standardize execution, but initiative-wide interoperability additionally requires shared or mappable data structures, stable identifiers, portable components, consistently described artifacts, and explicit specifications for transfers between models. Ontologies and KG can enrich such specifications with machine-interpretable scientific context. Their detailed alignment and validation, however, remain complementary semantic tasks and should not be conflated with the technical readiness of the workflow pipeline itself. The technical, semantic, and conceptual layers of interoperability are described in greater depth by Janssen et al. [8].

3.3 | Extending the Scope to MaterialsCommons: Dynamic Workflow Orchestration With PerQueue

The activities described in this article are rooted in MD but are intended to contribute to a broader, internationally interoperable workflow ecosystem. MaterialsCommons places workflows at the center of this extension: its objective is to connect established workflow environments and their tools, make workflow nodes FAIR and machine-readable, and ultimately enable workflows defined in different environments to be combined [7]. Among the frameworks contributing to this effort are AiiDA [63] and PerQueue [64]. PerQueue is considered here as one concrete example of how capabilities developed outside the MD-supported environments can complement the wider workflow landscape.

Modern computational research increasingly relies on complex workflows that combine traditional simulations with data-driven methods such as ML interatomic potentials (MLIPs) and self-driving labs. Managing these workflows efficiently remains a challenge [65], which has driven the development of numerous workflow managers. PerQueue provides one example of how this infrastructure can support the next stage of adaptive materials research.

Put briefly, PerQueue is designed to orchestrate Python-based workflows on HPC clusters.

Figure 27 illustrates how 23 lines of Python can use PerQueue to orchestrate a complex dynamic workflow for active learning of machine-learning interatomic potentials. Central to PerQueue is the idea of representing workflows as directed acyclic graphs (DAGs), with the possibility of using PerQueue task abstractions to create the DAG dynamically at workflow runtime. PerQueue uses MyQueue [66] to easily integrate with most HPC clusters, making PerQueue workflows interoperable with a large number of underlying scheduling and hardware configurations.

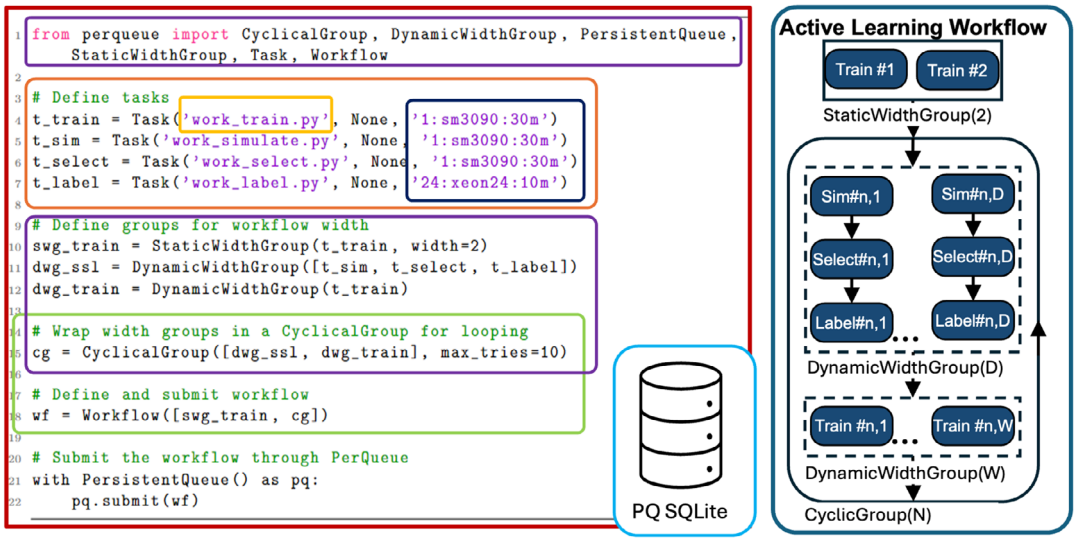


FIGURE 27 | Illustration showing how to go from a PerQueue Python script (red box) to a full active learning workflow (workflow on the right) by orchestration of smaller task-specific Python scripts (yellow box). The dependencies of each task are specified in advance using NetworkX (DAG) syntax (green box). PerQueue can orchestrate dynamic workflows with its implementation of various Task Groups (purple box). With PerQueue one can specify different resources for each task using MyQueue syntax (small dark blue box). In addition, each workflow writes to a SQLite database that contains data and metadata and handles the routing of data between dependent tasks (light blue box).

The principal direction for PerQueue's continued development is to leverage its dynamic workflow capabilities to accelerate scientific discovery through interdisciplinary workflows that span traditionally separated computational and experimental domains while at the same time making it AI-ready.

An ambitious target is the coupling of self-driving laboratories with ab initio calculations, where experimental measurements and simulations inform each other within a single orchestrated workflow, closing the loop between prediction and synthesis. Another goal is making PerQueue AI-ready by having a working integration with AI agents such that users of PerQueue can easily handle both experimental and computational metadata, working toward a PerQueue with agentic capabilities capable of doing material screenings autonomously. These directions are currently being pursued within the newly established Pioneer Center for Accelerating P2X Materials Discovery (CAPeX), which provides both the scientific use cases and the infrastructure to stress-test PerQueue on heterogeneous, multifidelity discovery campaigns.

3.4 | Priorities for Continued Development

AI-readiness should consequently be treated as a gradual and application-dependent profile rather than as a binary state or a universal target level. A laboratory data-acquisition pipeline, a high-performance simulation workflow, a multiscale modeling chain, and an AI-controlled production process have different technical requirements and do not need to converge on one architecture. The appropriate development path depends on the intended use, but recurring priorities include versioned data and metadata structures, reproducible execution environments, reusable artifact descriptions, explicit provenance, automated validation, and documented programmatic interfaces. These elements provide the basis for workflows that remain useful to domain experts while also becoming accessible to machine-learning systems and autonomous agents.

Acknowledgments

The authors thank the German Federal Ministry for Research, Technology and Space (BMFTR) for financial support of the project Innovation-Platform MaterialDigital through project funding Grant no. 13XP5094E (BAM), 13XP5094C (MPI SusMat), 13XP5195I (HSO DigiChrom), 13XP5226E (ICAMS) and 13XP5094A (KIT).

Open Access funding enabled and organized by Projekt DEAL.

Funding

This study was supported by the Bundesministerium für Bildung und Forschung (13XP5094A, 13XP5094C, 13XP5226E, and 13XP5094E).

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are published in public resources as stated at the respective places or available from the corresponding author upon reasonable request.

References

1. M. Schilling, P. von Hartrott, J. Waitelonis, et al., "Semantic Modeling in Materials Science and Engineering With Platform MaterialDigital Core Ontology 3.0," *Advanced Engineering Materials* n/a, no. n/a (2026): e71106.

2. C. R. C. Rêgo, J. Schaarschmidt, T. Schlöder, et al., "SimStack: An Intuitive Workflow Framework," *Frontiers in Materials* 9 (2022): 877597.

3. J. Janssen, S. Surendralal, Y. Lysogorskiy, et al., "Pyiron: An Integrated Development Environment for Computational Materials Science," *Computational Materials Science* 163 (2019): 24.

4. S. Bekemeier, C. R. Caldeira Rêgo, H. L. Mai, et al., "Advancing Digital Transformation in Material Science: The Role of Workflows Within the MaterialDigital Initiative," *Advanced Engineering Materials* 27, no. 8 (2025): 2402149.

5. Website MaterialDigital, Platform MaterialDigital, accessed May 09, 2026, https://www.material-digital.de.

6. J. Schaarschmidt, J. Yuan, T. Strunk, et al., "Workflow Engineering in Materials Design Within the Battery 2030+ Project," *Advanced Energy Materials* 12, no. 17 (2021): 2102638.

7. Website Materials Commons, Driving Digital Innovation in Materials Science within Europe, accessed August 09, 2026, https://www.materialscommons.eu.

8. J. Janssen, O. Waseda, B. Bayerlein, et al., "Supporting AI Readiness Through Digital Workflows in Materials Science," Advanced Engineering Materials (2026), manuscript submitted, https://doi.org/10.1002/adem.71262.

9. DigiMatUS Project, "CoatO: Coating Ontology, GitHub Repository," https://github.com/DigiMatUs/CoatO.

10. M. Glauer, R. West, S. Michie, and J. Hastings, "Esc-Rules: Explainable, Semantically Constrained Rule Sets," (2022), https://arxiv.org/abs/2208.12523.

11. J. Hastings, M. Glauer, R. West, et al., "Predicting Outcomes of Smoking Cessation Interventions in Novel Scenarios Using Ontology-informed, Interpretable Machine Learning," *Wellcome Open Research* 8 (2025): 503.

12. M. Glauer, M. Schilling, and M. Stappel, "A Practical Ontology Development Guide," https://github.com/scientific-ontology-network/ontology-development-guide/releases/download/v0.1.0/ontology-guide.pdf.

13. G. Zhu, J. Liao, G. Sun, and Q. Li, "Comparative Study on Metal/CFRP Hybrid Structures Under Static and Dynamic Loading," *International Journal of Impact Engineering* 141, no. 3 (2020): 103509.

14. G. Jeevi, S. K. Nayak, and M. Abdul Kader, "Review on Adhesive Joints and Their Application in Hybrid Composite Structures," *Journal of Adhesion Science and Technology* 33, no. 14 (2019): 1497.

15. F. Lambiase, S. I. Scipioni, C.-J. Lee, D.-C. Ko, and F. Liu, "A State-of-the-Art Review on Advanced Joining Processes for Metal-composite and Metal-Polymer Hybrid Structures," *Materials* 14, no. 8 (2021): 1890.

16. J. Zhou, X. Hong, and P. Jin, "Information Fusion for Multi-source Material Data: Progress and Challenges," *Applied Sciences* 9, no. 17 (2019): 3473.

17. HybridDigital Project, "HybridDigital Workflow Application, GitLab Repository," https://gitlab.cc-asp.fraunhofer.de/bjarsch/HybridDigital.

18. HybridDigital Project, "HybridDigital datasets, MaterialDigital Data Portal," https://dataportal.material-digital.de/organization/hybriddigital.

19. W. Brocks, D. Steglich, I. I. Milne, R. Ritchie, and B. Karihaloo, *Comprehensive Structural Integrity* (Pergamon, 2007), 107–136, ISBN 978-0-08-043749-1, https://www.sciencedirect.com/science/article/pii/B9780080437491003292.

20. E. B. Farahani, M. A. Eder, M. Alizadeh-Sh, et al., "An Experimentally Validated Thermomechanical Model for a Parametric Study on Reducing Residual Stress in Cast Iron Repair Welding," *The International Journal of Advanced Manufacturing Technology* 134 (2024): 5787.

21. E. Borzabadi Farahani, B. Sobhani Aragh, A. Sarhadi, and D. Juhre, "A framework to Model Thermomechanical Coupled of Fracture and Martensite Transformation in Austenitic Microstructures," *Thin-Walled Structures* 183 (2023): 110435.

22. H. Ji, L. Fuhrmann, J. F. Meza Gonzalez, and F. Rhein, "Optimization-Based Framework for Kernel Parameter Identification in Multi-Material Population Balance Models," *Digital Chemical Engineering* 17 (2025): 100272.

23. T. Harth, S. Schwan, J. Lehn, and F. G. Kollmann, "Identification of Material Parameters for Inelastic Constitutive Models: Statistical Analysis and Design of Experiments," *International Journal of Plasticity* 20, no. 8-9 (2004): 1403.

24. J. Tong, Z.-L. Zhan, and B. Vermeulen, "Modelling of Cyclic Plasticity and Viscoplasticity of a Nickel-Based Alloy Using Chaboche Constitutive Equations," *International Journal of Fatigue* 26, no. 8 (2004): 829.

25. S. Singer and J. Nelder, "Nelder-Mead Algorithm," *Scholarpedia* 4, no. 7 (2009): 2928.

26. R. Storn and K. Price, "Differential Evolution–a Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces," *Journal of Global Optimization* 11, no. 4 (1997): 341.

27. P. Virtanen, R. Gommers, T. E. Oliphant, et al, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods* 17 (2020): 261.

28. J. Janssen, J. George, J. Geiger, et al., "A Python Workflow Definition for Computational Materials Design," *Digital Discovery* 4, no. 11 (2025): 3149.

29. S. Issler, et al., *Abschlussbericht AiF-Vorhaben 10781* (MPA, Universität Stuttgart, 2000), Vol. 10781.

30. B. Böttger, M. Apel, M. Budnitzki, J. Eiken, G. Laschet, and B. Zhou, "Calphad Coupled Phase-Field Model With Mechano-Chemical Contributions and its Application to Rafting of γ'in CMSX-4," *Computational Materials Science* 184 (2020): 109909.

31. MICRESS, "MicPy: Python Tools for Reading and Processing MICRESS Output Files, Software Documentation," https://docs.micress.de/micpy.

32. L. Koschmieder, "MICRESS Pyiron Workflow, MaterialDigital Workflow Store," https://workflows.material-digital.de/workflow/95.

33. L. Koschmieder, "MICRESS Simstack Workflow, MaterialDigital Workflow Store," https://workflows.material-digital.de/workflow/105.

34. L. Koschmieder, "MICRESS-EarlyStageGrowth Workflow Active Node, GitHub Repository," https://github.com/lukas-koschmieder/MICRESS-EarlyStageGrowth.

35. L. Koschmieder, "MICRESS-PlotPhaseFractions Workflow Active Node, GitHub Repository," https://github.com/lukas-koschmieder/MICRESS-PlotPhaseFractions.

36. T. Mori and K. Tanaka, "Average Stress in Matrix and Average Elastic Energy of Materials With Misfitting Inclusions," *Acta Metallurgica* 21 (1973): 571.

37. Z. Hashin and S. Shtrikman, "A Variational Approach to the Theory of the Effective Magnetic Permeability of Multiphase Materials," *Journal of Applied Physics* 33 (1962): 3125.

38. Z. Hashin and S. Shtrikman, "A Variational Approach to the Theory of the Elastic Behaviour of Multiphase Materials," *Journal of the Mechanics and Physics of Solids* 11 (1963): 127.

39. J. C. Halpin, "Effects of Environmental Factors on Composite Materials," (1969), Semantic Scholar Corpus ID: 135872649..

40. S. Advani and C. Tucker, "The Use of Tensors to Describe and Predict Fiber Orientation in Short Fiber Composites," *Journal of Rheology* 31 (1987): 751.

41. International Organization for Standardization, "ISO/FDIS 23726-3: Automation systems and integration—ontology-based interoperability—part 3: Industrial data ontology, ISO," (2026), https://www.iso.org/standard/87560.html, Final Draft International Standard.

42. MaterialDigital, "PMD Core Ontology, Project documentation," https://materialdigital.github.io/core-ontology/.

43. OTTR Project, "Reasonable Ontology Templates, Project Website," https://ottr.xyz/.

44. "Micromechanics Plugin for Abaqus," accessed December 16, 2025, https://3dswym.3dexperience.3ds.com/en/post/simulia-community/micromechanics-plugin-for-abaqus_4167.

45. A. Ispas, A. Bund, T. Sörgel, et al., "The Need for Digitalisation in Electroplating - How Digital Approaches can Help to Optimize the Electrodeposition of Chromium from Trivalent Electrolytes," *Journal for Electrochemistry and Plating Technology* (2021).

46. J. Harter, E. Garcia Trelles, C. Schweizer, et al., "DigiChrom: A Domain Ontology for Semantic Representation of Trivalent Chromium Platings and Its Large Language Model-Based Alignment With Multiple Mid-Level Ontologies," *Advanced Engineering Materials* (2026): e71085.

47. M. Tandler, T. Seifert, and R. Mohrmann, "Bestimmung Von Spannungs-Dehnungskurven bei erhöhter Temperatur aus registrierenden Eindruckversuchen," in (Tagung Werkstoffprüfung, 2006).

48. H. M. Sajjad, T. Chudoba, and A. Hartmaier, "Inverse Method to Determine Parameters for Time-Dependent and Cyclic Plastic Material Behavior from Instrumented Indentation Tests," *Materials* 17, no. 16 (2024): 3938.

49. M. Sebastiani, K. Johanns, E. Herbert, and G. Pharr, "Measurement of Fracture Toughness by Nanoindentation Methods: Recent Advances and Future Challenges," *Current Opinion in Solid State and Materials Science* 19, no. 6 (2015): 324, recent Advances in Nanoindentation.

50. F. Teichert, F. Fuchs, P. Schulze, et al., "Submitted to Advanced Engineering Materials (Manuskript-ID: 4221718)," 2026.

51. F. Fuchs, F. Teichert, and P. Schulze, "Dimograph Active Learning of Gaussian Process Regression," https://workflows.material-digital.de/workflow/127.

52. L. Rizzi, A. Zienert, J. Schuster, M. Köhne, and S. E. Schulz, "Electrical Conductivity Modeling of Graphene-Based Conductor Materials," *ACS Applied Materials & Interfaces* 10, no. 49 (2018): 43088.

53. L. Rizzi, A. Zienert, J. Schuster, M. Köhne, and S. E. Schulz, "Computationally Efficient Simulation Method for Conductivity Modeling of 2D-Based Conductors," *Computational Materials Science* 161 (2019): 364.

54. L. Niemann, F. Fuchs, M. Gruschwitz, et al., "Towards Lightweight Conductors: Improving the Conductivity in Graphitic Films by Transition Metal Additives," *Diamond and Related Materials* 147 (2024): 111310.

55. M. Stevens, F. Fuchs, A. Hermannsdorfer, et al., "Wafer Level Drop Casting and $AlCl_3$ Doping for Highly Conductive Thin Graphene Paths," *Diamond and Related Materials* 153 (2025): 111989.

56. F. Teichert, "DiMoGraph Ontology for Graphene-Based Materials," https://gitlab.com/f.teichert/dimograph-ontology.

57. F. Teichert, "DiMoGraph Ontology for Graphene-Based Materials," https://doi.org/10.5281/zenodo.21263438.

58. I. Malashin, V. Tynchenko, A. Gantimurov, V. Nelyub, and A. Borodulin, "Applications of Long Short-term Memory (LSTM) Networks in Polymeric Sciences: A Review," *Polymers* 16, no. 18 (2024): 2607.

59. M. Baratchi, C. Wang, S. Limmer, et al., "Automated Machine Learning: Past, Present and Future: M. Baratchi et al.," *Artificial Intelligence Review* 57 (2024): 122.

60. J. Janssen, M. G. Taylor, P. Yang, J. Neugebauer, and D. Perez, "Executorlib – Up-Scaling Python Workflows for Hierarchical Heterogenous High-Performance Computing," *Journal of Open Source Software* 10 (2025): 108–7782.

61. A. P. Thompson, H. M. Aktulga, R. Berger, et al., "LAMMPS - A Flexible Simulation Tool for Particle-Based Materials Modeling at the Atomic, Meso, and Continuum Scales," *Computer Physics Communications* 271 (2022): 108171.

62. A. Atila, M. Sadowski, J. Janssen, L. Talirz, andT. Hickel, glasagent/amorphouspy (2026), https://github.com/glasagent/amorphouspy-https://github.com/glasagent/amorphouspy.

63. S. P. Huber, S. Zoupanos, M. Uhrin, et al., "AiiDA 1.0, A Scalable Computational Infrastructure for Automated Reproducible Workflows and Data Provenance," *Scientific Data* 7 (2020): 300.

64. B. H. Sjølin, W. S. Hansen, A. A. Morin-Martinez, et al., "PerQueue: Managing Complex and Dynamic Workflows," *Digital Discovery* 3 (2024): 1832.

65. S. K. Steensen, P. W. V. Butler, J. R. Pedersen, et al., "The Necessity of Dynamic Workflow Managers for Advancing Self-Driving Labs and Optimizers," *Advanced Intelligent Discovery* 2 (2026): 202500067.

66. J. J. Mortensen, M. Gjerding, and K. S. Thygesen, "MyQueue: Task and Workflow Scheduling System," *Journal of Open Source Software* 5, no. 45 (2020): 1844.

Supporting Information

Additional supporting information can be found online in the Supporting Information section. Supporting Information contains Table S1, which reports the explicit numerical levels underlying the color-coded AI-readiness profiles in Table 1, and Table S2, which summarizes the practical DiMad comparison of pyiron and SimStack for MICRESS integration.