Research Article | Open Access

Journal of Materials Informatics

Jiang *et al. J. Mater. Inf.* **2026**, *6*, 59 **DOI:**10.20517/jmi.2026.64

OFD: an Open Format Data schema for mathematical models and problem-related parameterization in materials science

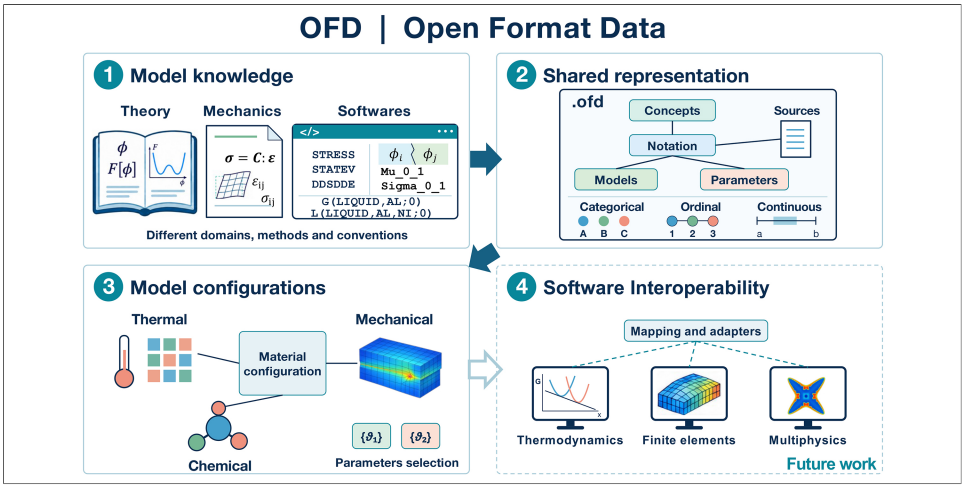
Yuxun Jiang[1,2,*], **Dhanunjaya K. Nerella**[1], **Ingo Steinbach**[1,2]

Keywords:
Computational materials science, knowledge representation, FAIR, JSON schema

Citation: Jiang, Y.; Nerella, D. K.; Steinbach, I. OFD: an Open Format Data schema for mathematical models and problem-related parameterization in materials science. *J. Mater. Inf.* **2026**, *6*, 59. https://dx.doi.org/10.20517/jmi.2026.64

Received: 29 Jul 2026
First Decision: 27 Aug 2026
Revised: 15 Oct 2026
Accepted: 28 Oct 2026
Published: 27 Nov 2026

Academic Editor:
Ankit Agrawal
Copy Editor:
Pei-Yun Wang
Production Editor:
Pei-Yun Wang



Abstract

Here we present the Open Format Data (OFD) schema, a JavaScript Object Notation (JSON) schema specification offering a unified, machine-readable format for storing "models" - a set of mathematical expressions mapping functional variables in an arbitrary configurational space onto each other - and "Data" - parameters relating the mathematical problem to a real-world problem - in one self-contained file. The schema organizes all information into four representative layers accompanied by metadata, each handling a distinct part of the model and the data. The nomenclature layer declares physical quantities, source-specific notations, and generalized coordinate systems used throughout the file. The model layer represents equations as symbolic patterns, linking each one explicitly to entries in the nomenclature layer. The parameter layer stores numerical values and analytical expressions, localizing them through coordinate selection. Finally, the material layer assembles models and parameters into coherent descriptions of specific problems in material systems, on which case we restrict ourselves. The schema reaches domain generality through a dimension-domain abstraction. The format is designed to align with Findable, Accessible, Interoperable, and Reusable (FAIR) principles and to support the progressive adoption of semantic technologies without disrupting existing JSON-based workflows.

Check for updates

[1]Interdisciplinary Centre for Advanced Materials Simulation, Ruhr-Universität Bochum, Bochum 44801, Germany.
[2]OpenPhase Solutions GmbH, Bochum 44799, Germany.

*Correspondence to: Yuxun Jiang, Interdisciplinary Centre for Advanced Materials Simulation, Ruhr-Universität Bochum, 44801, Germany. E-mail: yuxun.jiang@rub.de

INTRODUCTION

Computer simulations have become central to scientific research across disciplines, from physics and chemistry to engineering, biology, finance, and the social sciences. As scientific problems have grown more complex, the need for coupled modeling strategies has increased as well, especially for multi-scale and multi-physics interactions that cannot be captured by isolated single-scale descriptions[1]. This requirement is particularly prominent in materials science, where advanced modeling approaches like phase-field simulations and crystal plasticity models usually incorporate thermodynamics, elasticity, fluid flow, and plastic deformation within a single framework. Such coupling is essential because single-mechanism models cannot capture how physical processes interact during microstructural evolution[2,3]. The same trend is reflected in initiatives such as Integrated Computational Materials Engineering[4] and in software platforms for large-scale multi-physics simulation[5-8]. However, these integrated simulation workflows are also experiencing a major challenge: while the reliability of simulation depends on accurate data and parameter transfer across scales, different disciplines often rely on incompatible symbols, units, and formalisms for equivalent quantities. This makes model reuse and cross-disciplinary simulation difficult. Recent work on Findable, Accessible, Interoperable, and Reusable (FAIR) and open materials research highlights the importance of shared metadata and accessible computational context for data reuse[9-11]. To make this issue concrete, we focus on two representative subfields in computational materials science, crystal plasticity modeling and the CALculation of PHAse Diagrams (CALPHAD) method.

Crystal plasticity modeling is a key tool for studying the mechanical behavior of crystalline materials at the grain and sub-grain scale. It is based on the assumption that each grain can be treated as a continuum and that macroscopic deformation emerges from the collective activity of slip systems within individual grains[3]. Over the past few decades, the method has evolved from simple phenomenological descriptions of single-crystal slip into physics-based frameworks that can capture dislocation dynamics, deformation twinning, martensitic transformations, and grain interactions in multiphase polycrystals under complex loading conditions[3,6]. This development reflects a broader trend in computational science, where increasingly detailed models are needed to describe coupled behavior across multiple scales. At the same time, crystal plasticity still lacks a standardized format for model and parameter exchange. Simulation platforms such as DAMASK, Abaqus, and OpenPhase each use their own conventions for units, symbols, constitutive equations, and parameter naming, which makes model transfer and reproducibility difficult[6,8,12]. For instance, a reference slip rate may be named shrt0 in one UMAT implementation (UMAT is a Fortran-based subroutine in Abaqus/Standard that allows you to define custom material constitutive models[12]) and sliprate_0 in another phase-field code. Elastic stiffness tensors alternate between full fourth-order C_{ijkl} notation, 6 × 6 Voigt matrices, and engineering constants (E, ν, G) depending on the application context. Although tools such as DAMASK provide considerable flexibility by supporting multiple constitutive laws and interfacing with different solvers[6], this flexibility also highlights the fragmentation of the ecosystem. Equivalent quantities may be encoded differently across platforms, and input parameters often lose their physical meaning outside a specific implementation. Ad hoc wrappers can connect one software pair at a time, but they do not solve the broader interoperability problem. This makes crystal plasticity a clear example of the larger issue discussed above and motivates the need for a common intermediate representation that preserves the semantic identity of model components across software platforms.

In contrast to crystal plasticity, the CALPHAD community has achieved a notable degree of standardization through the TDB file format[13]. TDB represents Gibbs energy as a function of temperature, composition, and, in the Compound Energy Formalism, sublattice site fractions, using a unified parameter syntax such as

PARAMETER G(PHASE, ...) that supports exchange across major software platforms including Thermo-Calc[13], PANDAT[14], OpenCalphad[15], and PyCalphad[16]. Similar syntax is also used for diffusion mobility data through MOBILITY parameters, which enables kinetic simulations in tools such as DICTRA[13]. This standardization has made TDB a foundation for CALPHAD databases and ICME workflows[4], but the format also constrains the introduction of new physical models because thermodynamic quantities must ultimately be expressed as polynomials or simple exponential functions, and the format does not natively support integrals or conditional logic. This becomes a limitation for third-generation CALPHAD models, especially those that aim to remain physically sound down to 0 K. The Debye model, for example, cannot be represented directly in TDB because it does not have a closed analytical solution[17], and similar difficulties arise for the Modified Quasichemical Model, which requires data structures not directly supported by standard TDB syntax[16,18]. As a result, the format achieves syntactic standardization, but not full semantic interoperability, and its limited extensibility remains a practical barrier to more advanced thermodynamic models.

The two subfields discussed above highlight the same structural problem from opposite directions. Crystal plasticity lacks a standardized exchange format, whereas CALPHAD relies on a standard that is too rigid for some of the models the field now needs. This problem becomes especially visible in coupled multiphysics simulations, such as phase-field studies of microstructure evolution, where thermodynamic driving forces from CALPHAD must be linked with plastic deformation responses from crystal plasticity. In practice, researchers still have to reconcile differences in units, symbols, model assumptions, and parameter semantics across communities, which makes coupling difficult and non-feasible to reproduce. Several approaches have been developed to address different parts of this problem. XTDB[19] provides a structured representation of CALPHAD database information, but offers limited flexibility for representing models beyond its intended scope. Ontology-based systems such as OSMO[20], NOMAD[21], OPTIMADE[22], PMDco[23], and the European Materials and Modelling Ontology (EMMO)[24] provide principled frameworks for machine-readable knowledge representation using resource description framework (RDF) and web ontology language (OWL) formalisms, which can increase the implementation barrier. Consequently, while these approaches provide valuable infrastructure for semantic interoperability, they do not directly solve the need for a lightweight and flexible format for representing computational models.

This preceding analysis highlights a structural gap in computational materials science, and similar issues are evident across many other areas of computational science. Across those research fields, empirical and semi-empirical models share a common fundamental structure in that they decompose into expression trees with parameters, but each subfield has accumulated its own dialect of symbols, units, and constitutive formalisms. This makes model representation, exchange, and reuse difficult across disciplines. More broadly, it creates a barrier to communication between scientific communities that could otherwise benefit from shared computational abstractions.

To bridge this gap, we propose the Open Format Data (OFD) schema, a JavaScript Object Notation (JSON) schema specification that provides a unified, self-contained format for storing both the mathematical structure of computational materials models and their instantiated parameter values. Although the schema is motivated here primarily through examples from computational materials science, its underlying design is intended to support a broader range of scientific models whose variables, parameters and scopes can be explicitly defined. The schema bears three principles: (i) strict separation of physical concepts from source-specific notation and numerical data (type–instance decoupling); (ii) representation of parameter scopes through generalized coordinate systems rather than domain-specific fields; and (iii) full machine readability via a formal JSON schema with validated identifier conventions. The format is intended to serve as a practical intermediate representation that is sufficiently human-readable for researchers to inspect and edit,

and highly organized for automated pipelines to parse and validate. Explicit definitions and reference relationships may also support mapping onto formal ontologies, although such mappings are out of the scope of this work. Through examples presented here, we illustrate a general approach to organize model knowledge that may help reduce communication barriers across scientific computing communities.

The remainder of this paper is organized as follows. MATERIALS AND METHODS presents the OFD schema specification, covering the nomenclature, model, parameter, and material layers. RESULTS AND DISCUSSION demonstrates how generalized coordinates can be applied across different domains through two illustrative examples involving CALPHAD and crystal plasticity models and parameters. A further phase field example shows how several physical contributions can be organized within a material configuration. The section then discusses design trade-offs, comparisons with other approaches and a proposed data curation workflow, followed by limitations and future developments.

MATERIALS AND METHODS

OFD framework

OFD records computational model knowledge as a set of linked entries in a JSON document. The schema follows JSON Schema Draft 2020-12 and specifies the permitted fields and value types. Four representative layers are designed to describe models and parameters, together with an additional metadata layer registering information about the file and its sources. Figure 1 summarizes the components and their references.

An OFD file comprises four representation layers: nomenclature, models, parameters, and materials, together with a metadata layer. Of these five top-level components, metadata, nomenclature, and models are required, while parameters and materials are optional. Models record equations whose symbols are linked to definitions in nomenclature, or to records imported explicitly from other OFD files. Each notation entry associates a symbol with a physical quantity or a model-specified concept. Within nomenclature, dimensions and domains are optional fields, and they are introduced only to locate parameter values through definition of the coordinate ranges. The parameters and materials arrays are also optional, allowing a model description to be recorded before a parameter set or material configuration is selected. Material configurations then combine selected models and appropriate parameter sets through explicit references, such that different combinations can be documented without duplicating their definitions. Lastly, OFD files can refer to one another through imports and exports optionally. A complete schema is included as Supplementary Materials.

All major objects in an OFD file are referenced by structured identifiers that must be unique within the file. The corresponding object types are indicated directly through the prefixes as summarized in Table 1. It is advised to keep the identifier names simple while the scientific contexts are documented in fields such as name, description, definition and latex. Users may adopt project-specific naming rules, including placeholders with object codes, such that the convention remains flexible for different data curation workflows.

Metadata layer

The top-level metadata object records the basic information of an OFD file. It requires name and date, while fields such as description, version, license and keywords are optional. Source records are stored in metadata.sources as an array. Each source entry requires an id and may include its type, title, DOI, URL, original citation text (raw) and notes (note).

As shown in Figure 2, notations and parameters reference a single source through src_ref, whereas models use an array of source identifiers. Dimensions and domains may also include source references. Multiple entries can refer to the same source record, preserving provenance at the entry level when information from

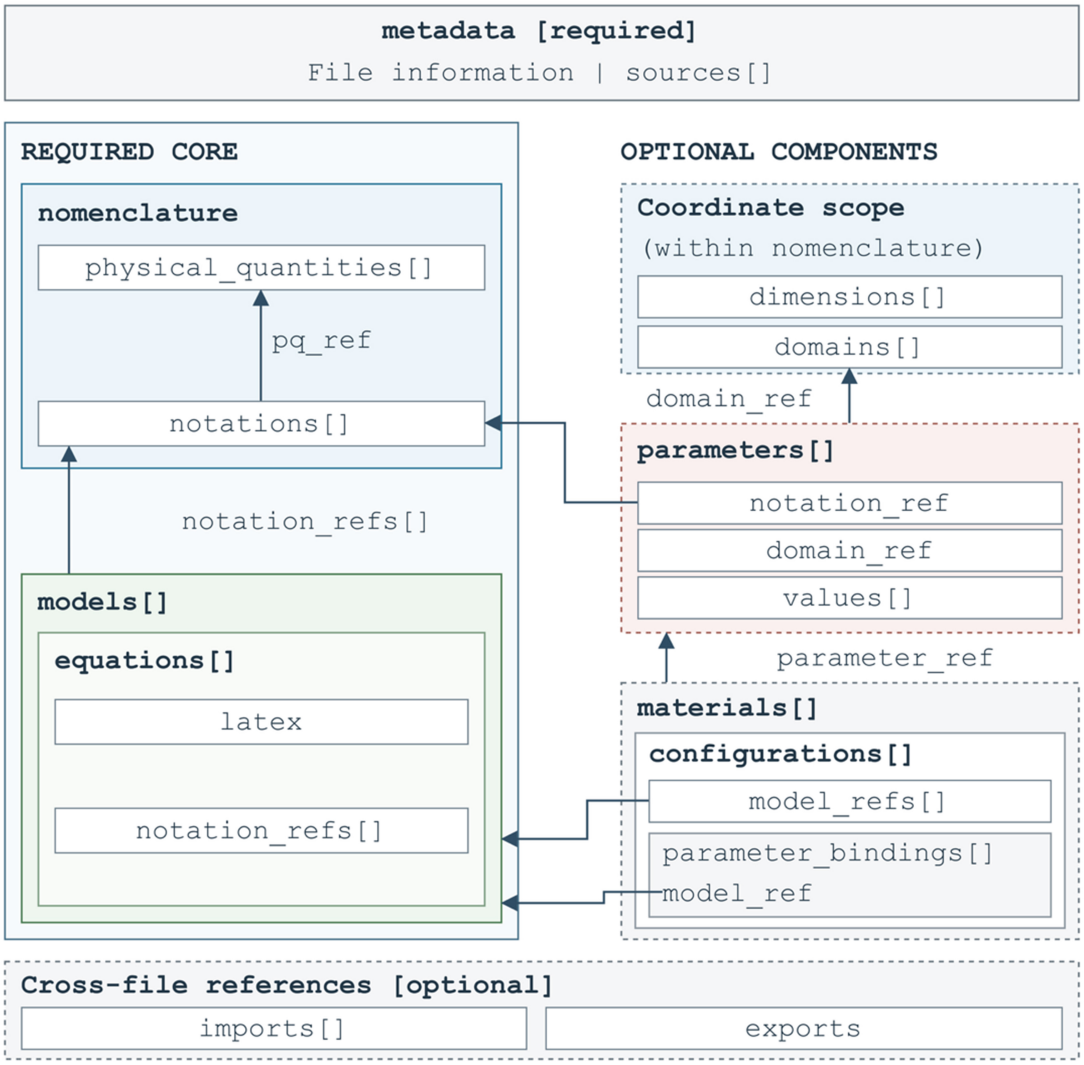


Figure 1. Overall structure of OFD. Arrows indicate selected references, while dashed outlines indicate optional components. OFD: Open Format Data.

Table 1. Identifier conventions used in the OFD schema

Entity	Prefix or pattern	Example
Source	src_<label>	src_du2003_alnbni
Physical quantity	pq_<label>	pq_gibbs_energy
Notation	<pq_id>.not_<label>	pq_gibbs_energy.not_G_m
Dimension	dim_<label>	dim_sublattice
Domain	dom_<label>	dom_AlNb2_occupancy
Model	mod_<label>	mod_regular_solution
Equation	<model_id>.eq_<label>	mod_regular_solution.eq_gibbs
Parameter instance	param_<label>	param_interaction_L_fcc_AlNi
Material	mat_<label>	mat_SAE4150
Material configuration	<material_id>.cfg_<label>	mat_SAE4150.cfg_A
Import	ofd_<label>	ofd_thermodynamic_database
External reference	<alias>:<object_id>	thermo:pq_gibbs_energy.not_G_m

Labels contain letters, digits, or underscores. External references use an alias declared in imports and are permitted only in fields that support them. The exports object lists existing identifiers and has no separate identifier. OFD: Open Format Data.

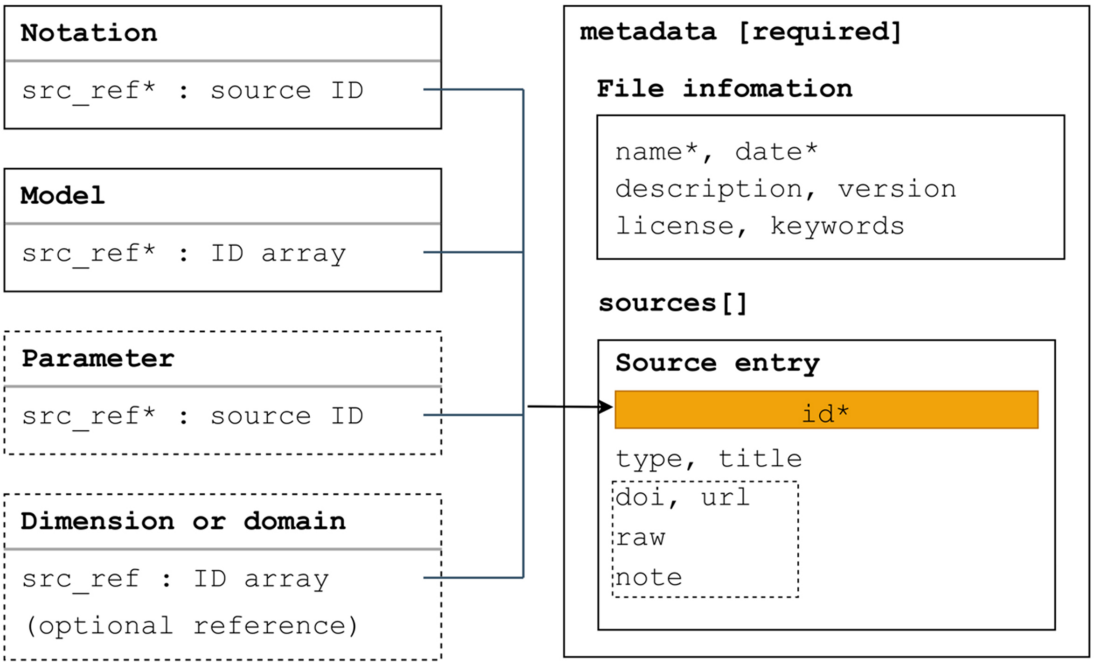


Figure 2. Metadata and source references in OFD. Arrows indicate references, dashed outlines mark optional components, while asterisks denote required fields. OFD: Open Format Data.

several sources is combined in one OFD file. The emphasis on explicit source and metadata records is consistent with FAIR data principles and work on machine-actionable metadata models[25].

Nomenclature layer

The nomenclature layer separates physical meaning from source-specific notation. It contains two required arrays, physical_quantities and notations. Optional dimensions and domains extend this representation to describe coordinates as shown in Figure 3.

By making this distinction explicit, OFD can represent the variation in symbols used across papers and software packages without losing the shared physical concept. For example, the Gibbs energy of a phase may appear as G, G_m, or with phase labels in superscripts and subscripts. OFD does not require these expressions to share a single canonical symbol; instead, it records the original notation and maps it to the corresponding physical quantity.

Physical quantities

The physical_quantities array records the concepts represented by model symbols, in which every entry contains an identifier and an optional description. In OFD, physical quantity is interpreted as a semantic category rather than as a strictly measurable physical variable. It may refer to a directly measurable quantity, a derived thermodynamic or mechanical quantity, an intermediate term introduced by a model, or a model-defined parameter family. These entries provide a common reference for notations that express the same concept. For example, different sources may denote Gibbs energy using different symbols or phase labels. Linking these notations to one physical quantity records their shared meaning while preserving the original expressions. More specific assumptions and definitions remain associated with the corresponding source notation.

{
"physical_quantities": [
{
"id": "pq_temperature",

"name": "temperature",
"description": "Thermodynamic temperature."
},
{
"id": "pq_gibbs_energy",
"name": "Gibbs energy",

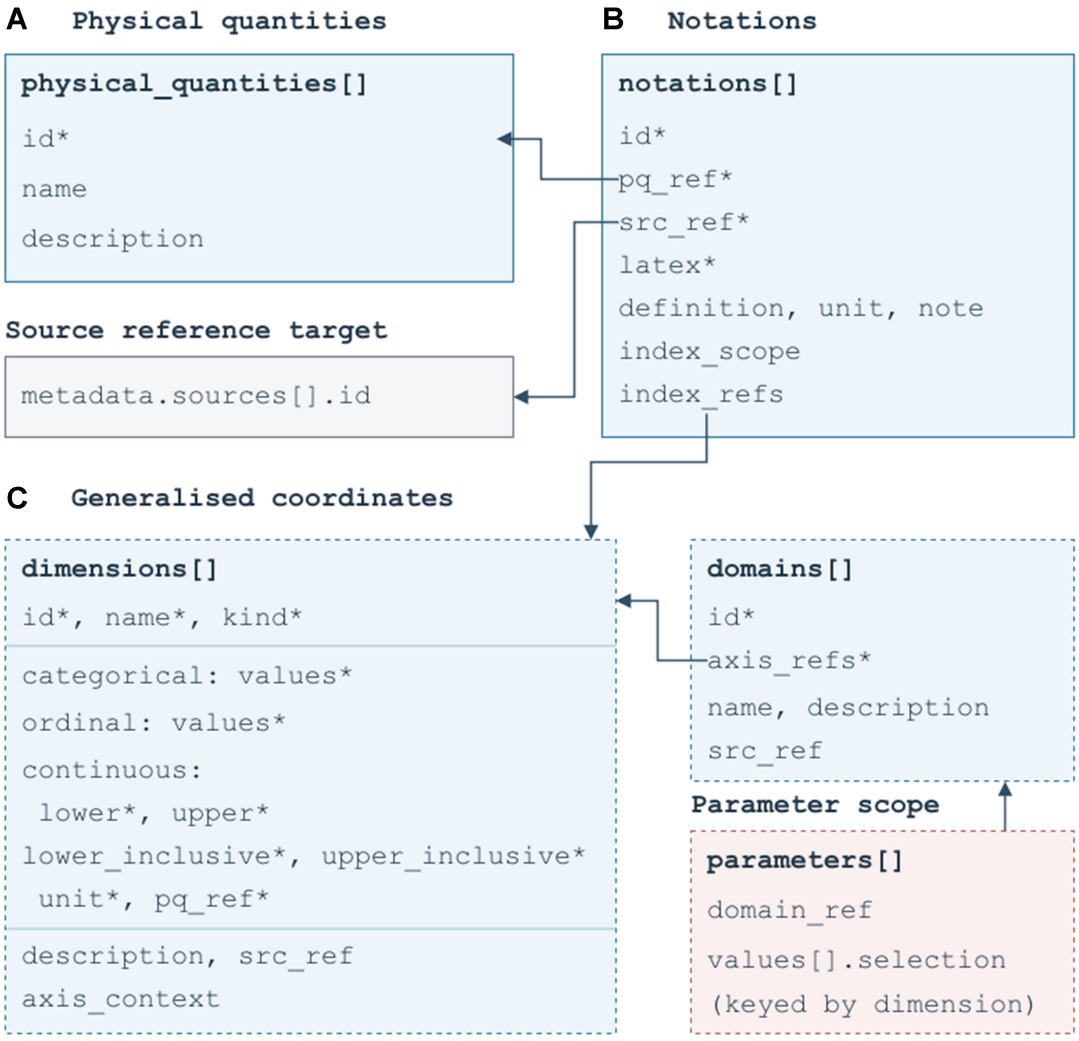


Figure 3. Structure of the OFD nomenclature layer: (A) physical quantities; (B) source notation; and (C) generalised coordinates. Arrows show selected references, dashed outlines mark optional components, and asterisks denote required fields. OFD: Open Format Data.

"description": "Molar Gibbs energy or related Gibbs energy expression."
},
{
"id": "pq_interaction_parameter",
"name": "interaction parameter",
"description": "Interaction parameter used in a thermodynamic model."
}
]
}

Notations

The notations array records the source-specific mathematical forms of physical_quantities. A mandatory pq_ref connects each notation to the corresponding physical quantity, so several symbols can share a common meaning. Equations and parameter instances then reference these notation entries, keeping both the original symbol and its definition accessible together.

For indexed expressions, index_scope can record the index meaning and permitted values as stated in its source, and index_refs links index symbols to explicitly defined dimensions. A notation therefore keeps its original mathematical form while its indices remain connected to the coordinate framework.


```
[{
"id": "pq_gibbs_energy.not_G_m_L",
"pq_ref": "pq_gibbs_energy",
"src_ref": "src_du2003_alnbni",
"latex": "{}^{\\mathrm{o}} G_{\\mathrm{m}}^{L}",
"unit": "J/mol",
"definition": "Molar Gibbs energy of the liquid phase."
},
{
"id": "pq_interaction_parameter.not_L_ijk",
"pq_ref": "pq_interaction_parameter",
"src_ref": "src_example",
"latex": "L_{i,j,k}",
"unit": "J/mol",
"definition": "Ternary interaction parameter.",
"index_scope": {
"i": {
"values": ["Al", "Nb", "Ni"],
"description": "Component index",
"evidence_text": "i, j, k denote components in the ternary system."
}
},
"note": "The source uses an abstract indexed form."
}
]
```


Generalized coordinates

The optional coordinate framework describes the scope of parameter values using dimensions and domains. It provides a common structure for distinctions such as chemical species, sublattice positions, slip systems, and temperature ranges. Categorical dimensions list distinct labels, such as chemical elements. Ordinal dimensions record ordered values, such as numbered sublattices. Both use a values array. Continuous dimensions specify lower and upper bounds, whether each bound is included, a unit, and a pq_ref linking the axis to its physical meaning.

A domain combines dimensions through its axis_refs array, whose order defines the domain axis order. The optional axis_context field links a dimension to a specific position on an ordinal axis, such as a numbered sublattice. Parameter instances refer to a domain through domain_ref, and each value record uses selection

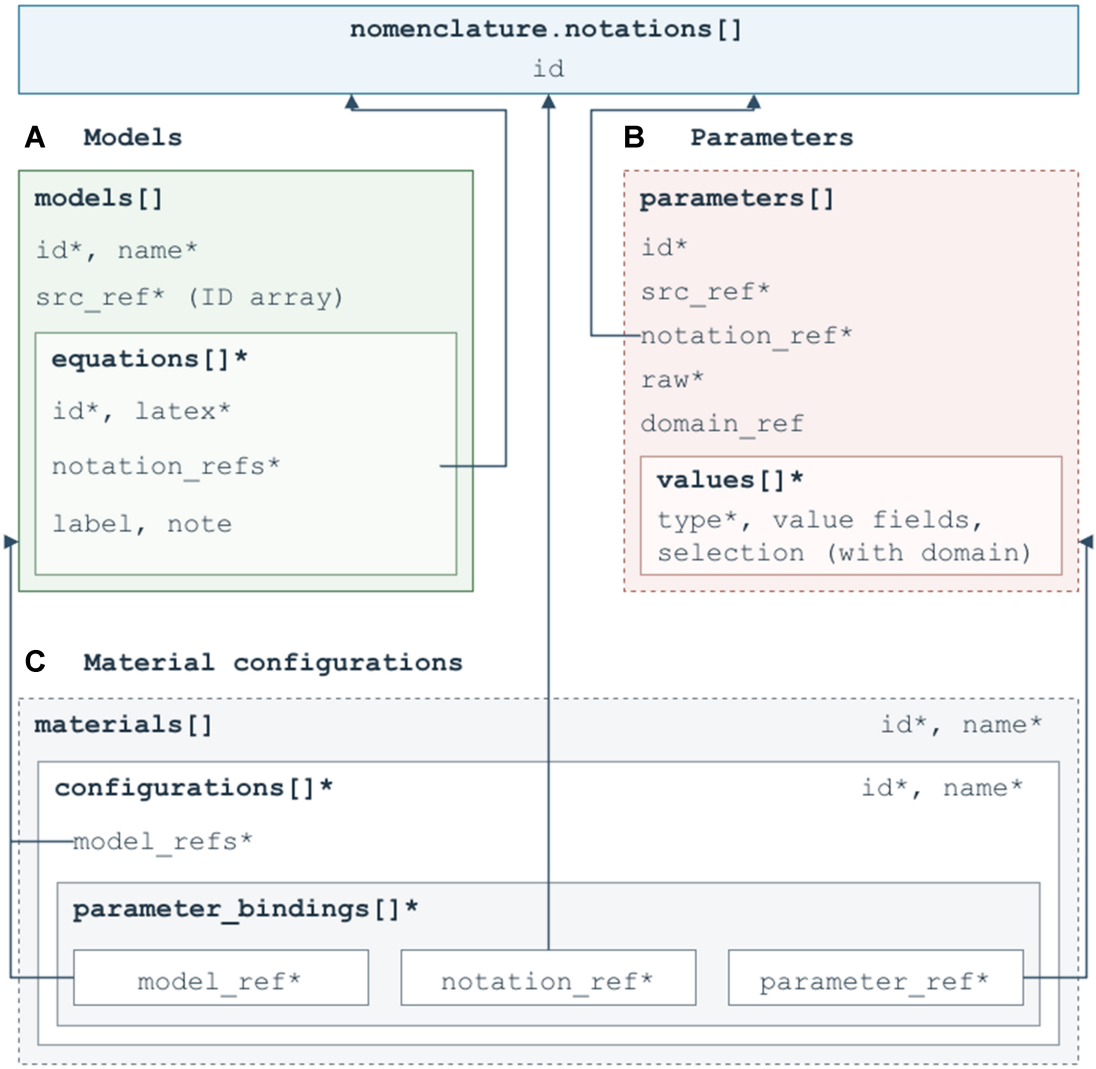


Figure 4. Models, parameters, and material configurations in OFD. (A) Models and their equations; (B) Parameter instances linked to source notation; (C) Material configurations that select models and bind parameter instances to model notations. Arrows show selected references between entries. Dashed outlines indicate optional components, and asterisks denote required fields. OFD: Open Format Data.

to identify its scope. Selections are keyed by dimension identifiers, so their meaning does not depend on the order of fields in the JSON object. For numerical arrays, data_axes specifies the nesting order from the outermost to the innermost axis, with each array extent matching the number of selected points on that axis. A combination associates a single coefficient with an ordered group of categorical labels, such as species contributing to a sublattice interaction. Values are recorded explicitly, without implicit broadcasting or interpolation.

Model, parameter and material layers

OFD stores model definitions, parameter instances, and material configurations in separate arrays linked by explicit references. Together, these components describe equations, available parameter data, and their intended combinations. The models array is required, while parameters and materials are optional. Figure 4 illustrates their relationships.

Each model contains one or more equations and identifies its sources through src_ref. An equation has an id, a latex expression, and a notation_refs array linking its symbols to notation records in the nomenclature layer or explicitly imported records. These links keep the expression and its definitions accessible together,

while the numerical data remain in separate parameter records. The notation reference identifies the source symbol and its associated concept, while the source reference records the provenance of the parameter instance. Several instances may refer to the same notation, allowing alternative values to be retained. Within values, each record specifies a type and stores numerical data, a LaTeX expression, or text. Additional fields can record units, independent variables, and notes. When a parameter includes a domain_ref, its value records use selection to specify where the values apply within the generalized coordinate framework. The optional materials array brings these records together for a material system or problem. Each material contains configurations in which model_refs selects models and parameter_bindings assigns parameter instances to their symbols.

The OFD examples were prepared manually from Markdown versions of publicly available papers. The Markdown content was manually checked against the source publications. Source records, model equations, and definitions of symbols and indices were recorded first. Available parameter values were linked to the corresponding notations and sources, retaining the stated units and conditions of use. Coordinate scopes and material configurations were added where needed. Missing information was left unresolved, and records lacking required information were treated as drafts. The completed files were parsed for valid JSON syntax and validated against the supplementary OFD schema using Python 3.12.11 and the Draft202012Validator class in jsonschema 4.25.1.

RESULTS AND DISCUSSION

The following examples illustrate how OFD represents model knowledge from different fields using a common structure. We first compare sublattice occupancy in CALPHAD with constituent and slip family descriptions in crystal plasticity to show how generalized coordinates locate parameter values. These examples then trace the links between model equations, source notation, and parameter instances. A final phase field example shows how several model contributions and their parameters can be assembled within a material configuration. Selected JSON snippets highlight the relevant fields. Complete OFD files, including the referenced definitions and source records, are provided as Supplementary Materials.

A common coordinate structure for different parameter scopes

Sublattice structure and constitution

The CALPHAD example uses selected $AlNb_2$ records from the Al Nb Ni assessment by Du *et al.*[26]. The source describes the phase using three sublattices, $(Al)_{4/15}(Nb)_{2/15}(Al,Nb,Ni)_{9/15}$ which indicates normalized sublattice site ratios of 4/15, 2/15, and 9/15, respectively. Each position has a categorical species axis, with Al on the first, Nb on the second, and Al, Nb, or Ni on the third. The field axis_context links each species axis to its sublattice position. The occupancy domain lists the three species axes in that order. Sublattice position supplies their context and does not add a fourth axis to the domain. This arrangement distinguishes an endmember from an interaction coefficient within the same domain. The Al:Nb:Ni endmember selects one species on each axis. For the Nb and Ni interaction on the third sublattice, the first two selections remain fixed, while the third uses a combination. The following fields belong to the interaction parameter record, and the relationship is sketched in Figure 5A.


```
{
 "domain_ref": "dom_AlNb2_occupancy",
 "values": [{
  "type": "numeric",
  "number": -15165,
  "unit": "J/mol",
  "selection": {
   "dim_species_s1": {"mode": "point", "value": "Al"},
   "dim_species_s2": {"mode": "point", "value": "Nb"},
```

```
    "dim_species_s3": {
      "mode": "combination",
      "values": ["Nb", "Ni"]
    }
   }
  }]
}
```

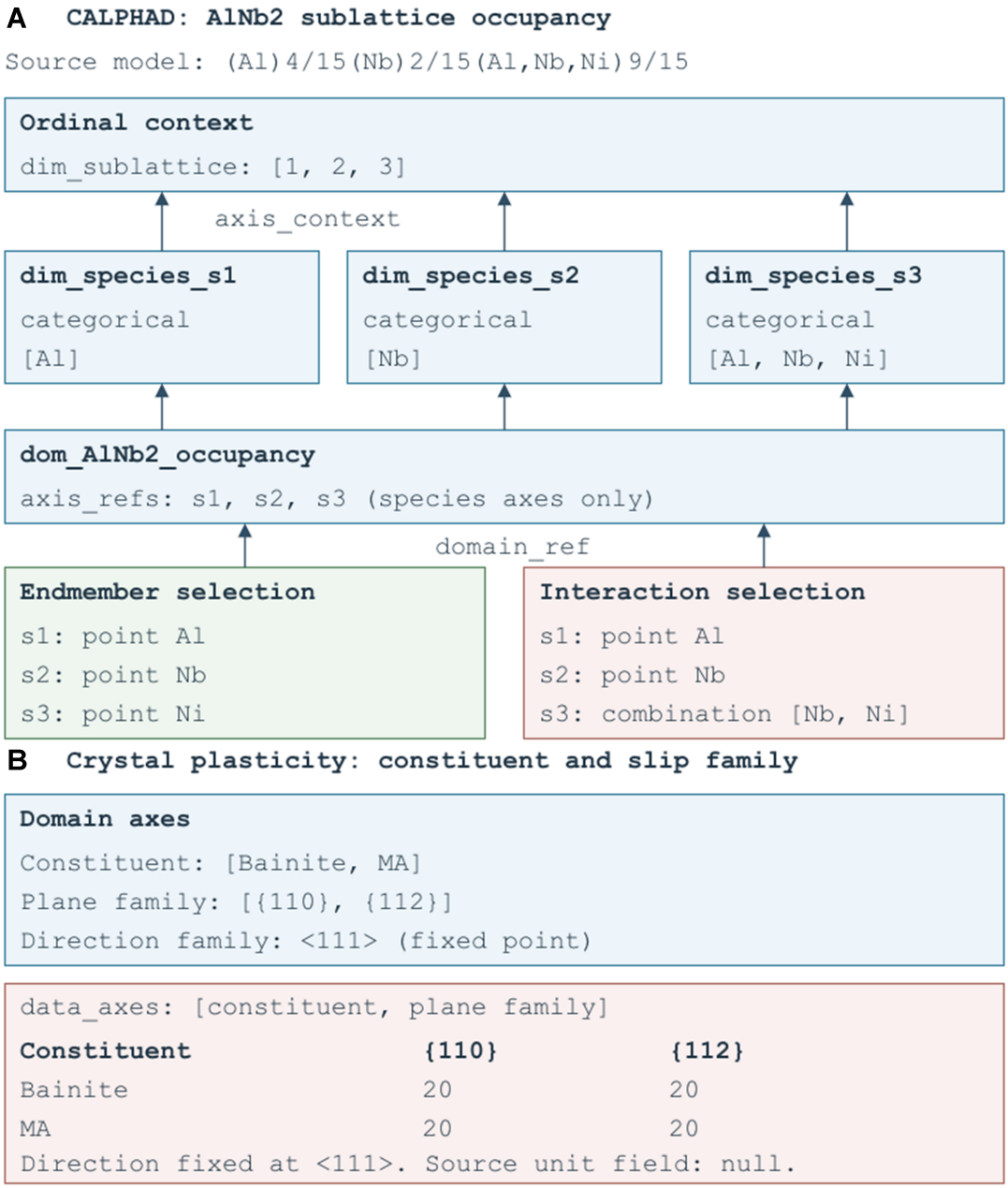



Figure 5. Generalised coordinates for parameter scopes in CALPHAD and crystal plasticity. (A) In CALPHAD, sublattice positions and constituent selections define the scope of endmember and interaction parameters; (B) In crystal plasticity, the scope of a parameter depends on the selected constituent and slip family. CALPHAD: CALculation of PHAse Diagrams.

Constituents and slip systems

In the crystal plasticity example, parameter scope follows a different organization. Fujita *et al.* describe bainitic steel using Bainite and MA constituents, with slip families involving the {110} and {112} planes and the <111> direction[27]. The OFD domain combines categorical axes for constituent, plane family, and direction family. MA retains its source meaning as a martensite and austenite constituent. These labels do not enumerate individual slip systems. The source provides a common rate exponent of 20 for each constituent and applies the slip relation to both families. Its OFD representation explicitly records a two-by-two array, in which rows follow the constituent axis and columns follow the plane family axis, as declared by data_axes. The direction is fixed by a point selection, as shown in Figure 5B.


```
{
  "domain_ref": "dom_bainite",
  "values": [{
    "type": "numeric",
    "data": [[20, 20], [20, 20]],
    "unit": null,
    "selection": {
      "dim_phase_bainite": {
        "mode": "points",
        "values": ["Bainite", "MA"]
      },
      "dim_plane_bainite": {
        "mode": "points",
        "values": ["{110}", "{112}"]
      },
      "dim_direction_bainite": {
        "mode": "point",
        "value": "<111>"
      }
    },
    "data_axes": ["dim_phase_bainite", "dim_plane_bainite"]
  }]
}
```


The repeated entries express the scope of the common source value. They are not four independent measurements, and no implicit broadcasting is needed to interpret the stored array. The source unit cell is blank, which is retained as null.

Together, these cases show how the same coordinate fields describe an ordered occupancy structure and a set of constituent and slip family labels. Their selection modes preserve a necessary distinction: a categorical combination identifies one interaction coefficient, whereas sampled coordinates locate separate array entries. Field-specific meanings remain in the axis definitions and model records.

Retaining model context around parameter values

Coordinates alone cannot explain the role of a parameter. The single model examples therefore link each value to source notation, a physical quantity, and the equation in which it appears. Figure 6 shows these links for the thermodynamic coefficient and the crystal plasticity rate exponent.

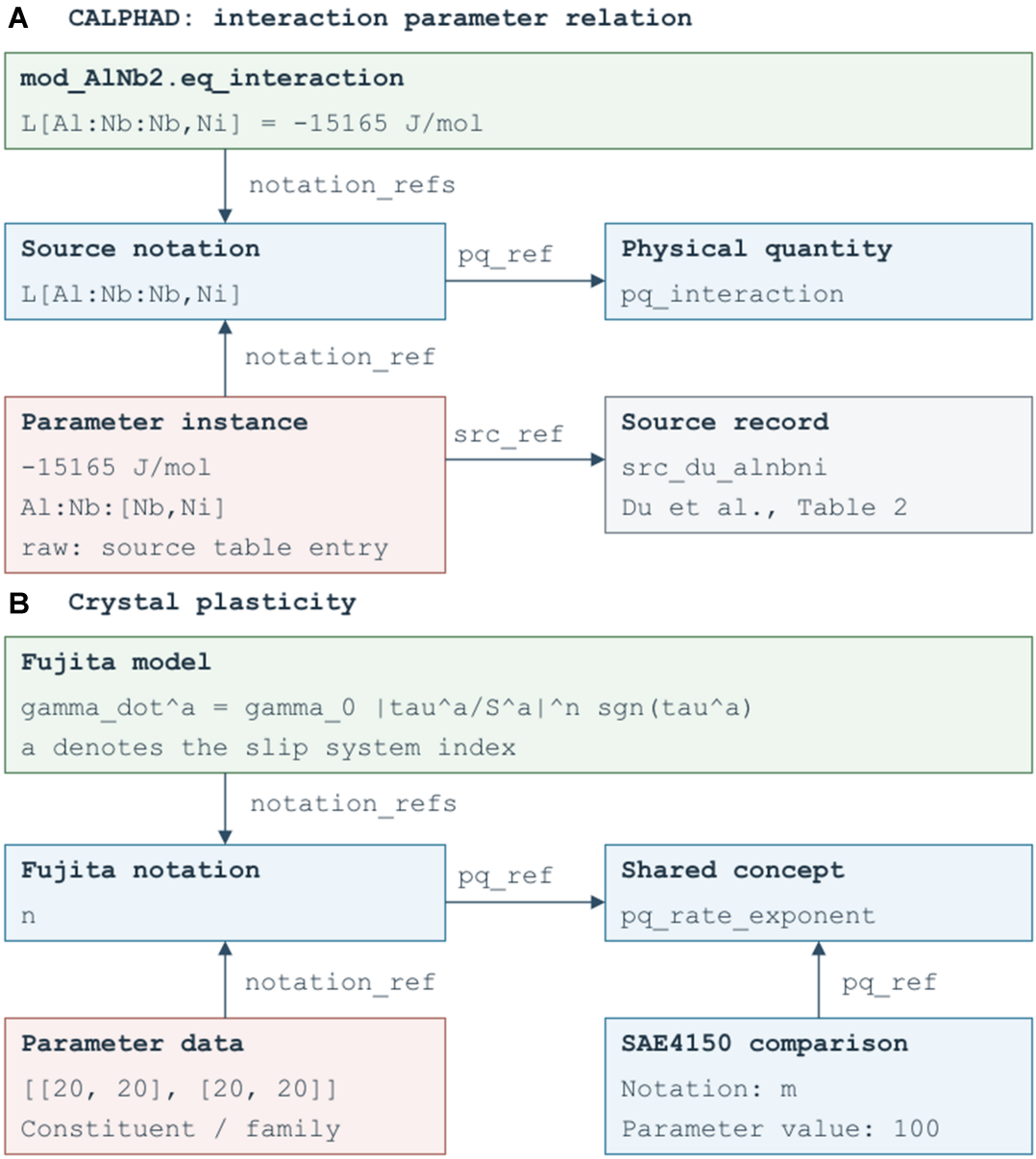


Figure 6. Links between equations, source notation, physical concepts, parameter values, and source records in (A) CALPHAD and (B) crystal plasticity. CALPHAD: CALculation of PHAse Diagrams.

For the CALPHAD interaction, pq_interaction records the concept and its notation preserves $L^{AlNb_2}_{Al:Nb:Nb,Ni}$. The equation references this notation through notation_refs. A separate parameter instance points to the same notation and contains the value of -15,165 J/mol with the occupancy selection described above. Its src_ref identifies the assessment, and raw retains the source table entry. A reader can thus follow the numerical value back to its expression, occupancy, and provenance.

The thermodynamic file also records an Al:Nb:Ni endmember offset of -45,000 J/mol relative to pure element reference terms. The source identifies this as a fictitious assigned value, and that qualification is retained. Both values use the source basis of a mole of atoms. Keeping the reference terms and source qualification alongside the offset avoids presenting it as an absolute Gibbs energy or a measured value. The example records the selected parameter relations without reconstructing missing reference functions.

For crystal plasticity, the same reference structure connects the exponent array to the slip rate equation and its source definitions. A second description from Schäfer *et al.* concerns martensitic SAE4150 steel[28]. Its slip relation includes back stress and uses a rate exponent denoted by *m*, with a value of 100. The two exponent notations refer to a shared physical quantity entry, while their equations, domains, values, and sources remain separate. This association makes their conceptual relationship visible without treating their constitutive descriptions as interchangeable.

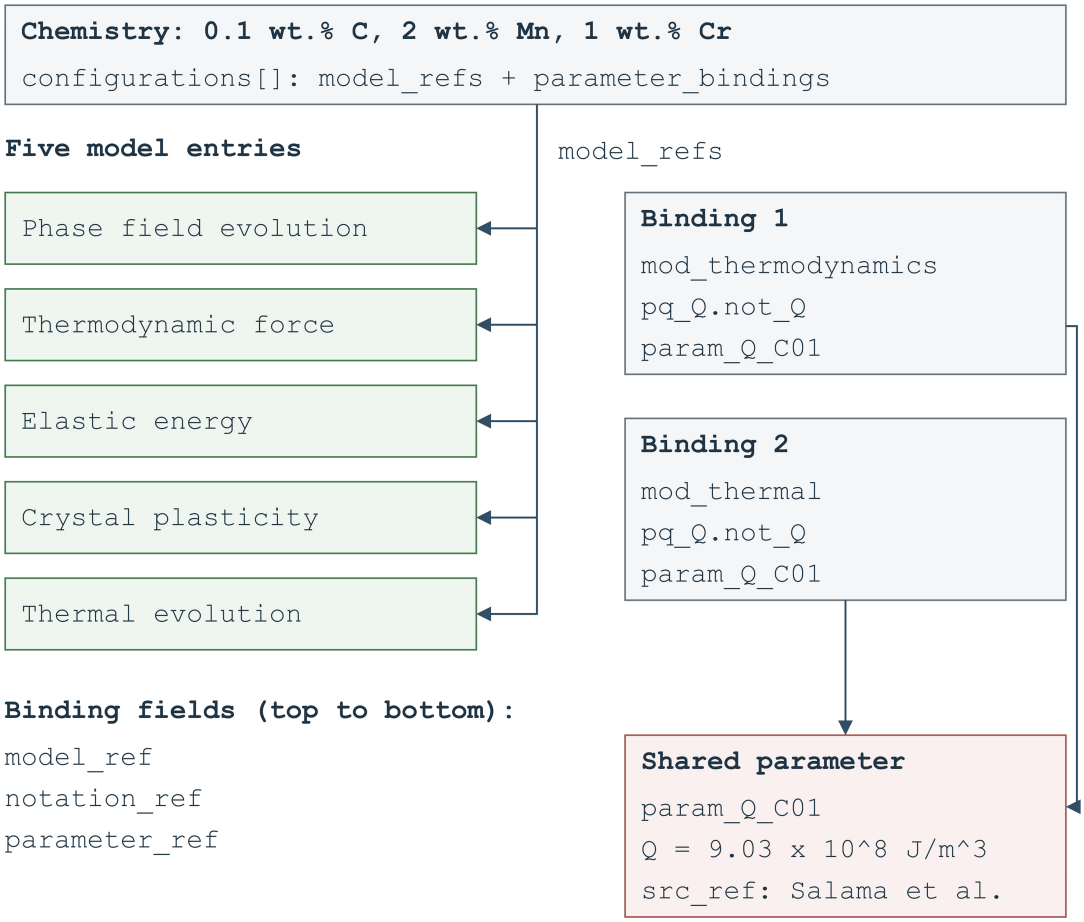
The SAE4150 source also supplies initial critical stresses of 187, 155, and 209 MPa for three alternative calibrations. OFD stores these as separate parameter instances associated with the same notation and coordinate scope. Each material configuration selects the instance belonging to one calibration. This separates two kinds of variation: coordinates describe where a value applies within a model, and configurations distinguish alternative parameter choices. The recorded subset includes the slip relations and selected parameters. The associated hardening evolution laws are outside its scope.

Assembling a multiphysics material configuration

The final example uses the steel transformation framework of Salama *et al.*[29] to show how these linked records are organized when several physical contributions belong to one problem, as present in Figure 7. The OFD file contains five model entries covering phase field evolution, thermodynamic driving force, elastic energy, crystal plasticity, and thermal evolution. In the source description, chemical and mechanical contributions enter the phase field formulation, plastic deformation affects the mechanical response, and temperature evolves with cooling and latent heat. Each contribution retains its own equations and notation references. A material configuration lists the five model entries and binds selected parameters to their notations.

Latent heat illustrates a parameter shared by two contributions. The thermodynamic driving force and thermal evolution both refer to the same recorded quantity *Q*. In the configuration for steel containing 0.1 wt.% C, two bindings select one parameter instance. Both references resolve to the same value and source record, so the configuration explicitly records the shared choice. The complete example contains three material entries with carbon contents of 0.1, 0.2, and 0.3 wt.%, each with 2 wt.% Mn and 1 wt.% Cr. Their configurations retain the same five model references and select the corresponding equilibrium temperature and latent heat instances. The remaining recorded bindings are unchanged. These entries illustrate how composition-dependent choices can be expressed without duplicating the model equations. The complete material object *mat_steel_C01* is provided in the Supplementary file *examples/phase_field_material_configuration/example.ofd.json*, including its configuration mat_steel_C01.cfg_martensite, selected models, and parameter bindings.


```
{
  "parameter_bindings": [
    {
      "model_ref": "mod_thermodynamics",
      "notation_ref": "pq_Q.not_Q",
      "parameter_ref": "param_Q_C01"
    },
    {
      "model_ref": "mod_thermal",
      "notation_ref": "pq_Q.not_Q",
      "parameter_ref": "param_Q_C01"
    }
  ]
}
```



C (wt.%)	T0 (°C)	Q (J/m^3)
0.1	717.5	9.03 x 10^8
0.2	678.3	8.41 x 10^8
0.3	638.9	8.71 x 10^8

Figure 7. Material configurations combining model contributions and selected parameter instances.

Across the examples, the coordinate structure identifies parameter scope, the notation links retain model context, and configurations record selected combinations. This progression shows how the same OFD components organize the specific source descriptions at increasing levels of complexity.

Main features and comparison with a predicate logic representation

OFD separates model definitions from parameter instances, allowing different parameter sets to refer to the same model without repeating the underlying equations. It also distinguishes physical concepts from source-specific notation, so records that use different symbols can still be related. These features make it possible to compare and select model components within material configurations. Moreover, mathematical expressions are retained in LaTeX so that they can be inspected directly. Generalized coordinates provide a common way to describe parameter scope when such a structure is needed. The example below compares this representation with an explicit logical description of the same domain information.

Within the CEF, the Gibbs free energy of a phase is generally expressed as a sum of contributions associated with different physical effects. Here, the excess Gibbs energy arising from interactions between constituents is used to illustrate the representation. According to Hillert[30], consider a substitutional phase φ with a sublattice structure $(A,B)_a(C,D)_b$, then the contribution from the interaction between A and B elements in the first sublattice with C occupying the second sublattice reads:

$$G^{\varphi}_{ex,A,B:C} = y_A^{(1)} y_B^{(1)} y_C^{(2)} \sum_{n=0}^{N} L^{\varphi,n}_{A,B:C} (y_A^{(1)} - y_B^{(1)})^n \tag{1}$$

Here, a and b are fixed site ratios, $y_s^{(i)}$ denotes a site fraction of element s on ith sublattice, and $L_{A,B:C}^{\varphi,n}$ is an interaction coefficient of order n. Readers are referred to Hillert[30] for further details. The information required to interpret each coefficient may also be written in terms of locally defined predicates. Let the predicate ParameterOf(p,ϕ) associates a parameter record p with phase φ, and its constitution can be expressed by HasConstituent(p,i,k,s) which identifies species s occupying position k on ith sublattice. For the elements occupation in the preceding binary example, the relevant constituent is thus expressed as HasConstituent(p,1,1,A), HasConstituent(p,1,2,B) and HasConstituent(p,2,1,C), respectively. Its polynomial order is given by HasOrder(p,n). Such that the following constraints can be established:

$$\forall p \ \mathrm{BinaryInteractionParameter}(p) \Rightarrow \exists! n \ \mathrm{HasOrder}(p, n) \tag{2}$$

and

$$\forall p, \varphi \ \mathrm{BinaryInteractionParameter}(p) \wedge \mathrm{ParameterOf}(p, \varphi) \Rightarrow \exists t \ [\mathrm{TermOf}(t, \varphi) \wedge \mathrm{ContributesTo}(p, t)] \tag{3}$$

where n ranges over nonnegative integers, TermOf identifies a term belonging to the model, and ContributesTo associates a parameter with that term.

OFD represents the same information through structured fields and linked entries rather than explicit predicates. Constituent combinations are recorded by selection, their positions by axis_context, and the relevant scope by the referenced domain (domain_ref). Other relationships, such as how a coefficient enters a model, are conveyed through notation references, definitions, and equations. The resulting record makes important contextual information explicit without requiring all scientific meaning to be formalized as logical axioms. Compared with formats whose meaning must be inferred from software-specific conventions such as TDB, OFD can be inspected and edited once the schema is understood, without requiring prior ontology construction or vocabulary alignment.

This design reflects a trade-off between the compactness of domain-specific conventions and the reuse of a common recording structure. In CALPHAD, established notation can encode parameter roles very efficiently, whereas OFD may require several entries to state the same context explicitly. The advantage of the latter structure is that it can also accommodate models with different indexing schemes. Sublattice occupancy and slip-family selection, for example, can both be described using dimensions, domains, axes, and selections rather than model-specific fields. However, these entries are unnecessary when the scope is already defined unambiguously by an established convention, such as Voigt notation with a specified component order. Predicate logic could represent these relationships as well; the distinction lies not in the expressive capability itself, but in OFD's use of a shared data structure intended for practical model representation and comparison. Neither an OFD record nor a predicate-based representation can determine the physical compatibility of models and parameter sets from their metadata structure, and depends on the underlying physical assumptions and must be assessed separately.

Data curation at scale and large language model assistance

Preparing the examples required tracing definitions and parameter conditions across the supplied documents. A table value may depend on an earlier definition, a reference state, or a footnote. As collections grow, recovering and checking these connections demands increasing manual effort. Large language model (LLM) assistance offers a promising direction for preparing records and locating evidence for review.

Applications of LLMs to structured scientific extraction[31,32] suggest specific ways to support this work. Given an equation and its surrounding text, a model could draft notation entries with supporting passages for each definition. For a parameter table, it could associate values with units, row and column labels, and conditions stated in captions or footnotes. These details would inform coordinate selections, with unclear assignments left for review.

Draft review could use targeted questions, following the checking approach explored in ChatExtract[33]. For example, "does a value apply to all slip families or only one, and does a definition retain the source's stated conditions?" Each suggested correction should identify the affected entry and supporting passage. Unsupported assignments would remain unresolved. Since LLMs can produce plausible errors[34], reviewers must check this evidence before accepting changes. Our corrected examples do not establish extraction accuracy or a reduction in review effort.

Confidential sources require appropriate access controls and data handling arrangements. Local models reduce external data transfer, but documents, prompts, and generated records still need protection. OFD does not depend on a particular model or deployment environment.

Perspectives and future work

The examples illustrate selected uses of OFD, leaving broader applicability and extraction accuracy unevaluated. Its semantic content is limited to recorded concepts, definitions, and references. LaTeX does not formally encode operator structures or execution rules, and complete notation links cannot guarantee mathematical completeness or correctness. These boundaries motivate three priorities for further work.

First, a reviewed reference collection should pair source passages with OFD records and documented mapping decisions. Including ambiguous and incomplete cases would help define evaluation criteria for coverage, correct values and relationships, and information that remains unresolved.

Second, this collection would provide a basis for comparing manual curation with LLM assistance. Evaluation should account for source fidelity, omissions, and the time needed to check and correct suggestions. Documenting model versions, source preparation, prompts, and available context would make comparisons between local and hosted models interpretable.

Third, conversion studies should trace whether parameter scopes, units, reference states, and assumptions retain their meaning in selected target formats and interfaces. Adapters would need to account for software conventions absent from OFD. Generating executable code also requires decisions about variable scope, numerical methods, and initial and boundary conditions, followed by separate tests against reference results. Extending these studies to executed workflows would require calculation histories beyond source attribution, for which RO-Crate or Workflow Run RO-Crate could provide complementary records[35,36].

These efforts would benefit from contributions across materials modeling communities[37], whose different conventions may expose gaps missed by the present examples. Shared mappings and conversion rules could support further use of OFD. Tools released with these contributions should document versions,

dependencies, and reuse conditions following the FAIR Principles for research software[38].

CONCLUSIONS

In this work, we developed OFD as a schema for the structured representation of mathematical notation and model parameters that constitute a complete computational model. A central design principle of OFD is the separation of the physical or model-defined meaning of a quantity from the notation used in a particular source. This approach enables equations to be recorded in a source-faithful form while maintaining explicit relationships among models, symbols, and parameter instances.

The schema is particularly suited to the organization and management of model parameter data. Rather than being stored as isolated numerical values, parameter records are linked to source-specific notations, physical or model-defined quantities, and provenance information. This structure facilitates inspections of internal referential integrity, including whether each notation reference resolves to a notation entry within the same OFD file. It also supports modular reuse, allowing model components to be represented in separate OFD records and combined through explicit reference relationships.

The sublattice comparison defines a limited logical interpretation of the coordinate records and clarifies its boundaries. Further formalization would be needed to support reasoning about information currently retained in descriptions and mathematical expressions. Taken together, OFD provides a lightweight and practical representation framework for organizing model definitions and parameter information as structured data. The framework is not intended to replace expert interpretation or domain-specific assessments. OFD aims to reduce dependency on implicit domain conventions by making model definitions and parameter choices available for inspection and further processing. Structural checks can support record preparation, while source fidelity and scientific compatibility still require review. Future work should develop reviewed reference datasets and evaluation criteria, assess AI-assisted extraction, and examine conversions between OFD and simulation software. These efforts would help establish the scope of OFD as a shared representation for computational model knowledge.

DECLARATIONS

Acknowledgments

The authors acknowledge the coordination of the TurboHyTec project by the Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR).

Authors' contributions

Methodology, software, investigation, data curation, visualization, writing and original draft: Jiang, Y.
Data curation, writing and original draft: Nerella, D. K.
Conceptualization, methodology, funding, resources, supervision, writing, review and editing: Steinbach, I.

Availability of data and materials

Some results supporting the study are presented in the Supplementary Materials. Other raw data that support the findings of this study are available from the corresponding author upon reasonable request.

AI and AI-assisted tools statement

The authors used ChatGPT (version 5.4, released 2026.03.05) for language polishing during manuscript preparation. AI assistance in the figure was limited to generating schematic illustrations of an FEM mesh and dendrites using OpenAI's image generation tool. All other visual elements were created by the authors. The authors reviewed the AI-generated text and take full responsibility for the scientific content and final manuscript.

Financial support and sponsorship

This work was supported by the German Federal Ministry for Economic Affairs and Climate Action

(BMWK) through the AG Turbo program (Grant No. 03EE5159F) within the joint research project TurboHyTec (Turbomachinery for Hydrogen Technologies), Work Package WP 4.10, Subproject 4.10d (Teilvorhaben 4.10d - Stabilität von IN718 bei hohen Betriebsdauern).

Conflicts of interest

Jiang, Y. and Steinbach, I. are affiliated with OpenPhase Solutions GmbH, while the other author has declared no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

© The Author(s) 2026.

Supplementary Materials

Supplementary Materials

REFERENCES

1. Weinan, E. *Principles of multiscale modeling*. Cambridge University Press; 2011. https://web.math.princeton.edu/~weinan/papers/weinan_book.pdf. (accessed 2026-11-05).
2. Steinbach, I. Phase-field models in materials science. *Modell. Simul. Mater. Sci. Eng.* **2009**, *17*, 073001. DOI
3. Roters, F.; Eisenlohr, P.; Hantcherli, L.; Tjahjanto, D.; Bieler, T.; Raabe, D. Overview of constitutive laws, kinematics, homogenization and multiscale methods in crystal plasticity finite-element modeling: theory, experiments, applications. *Acta. Mater.* **2010**, *58*, 1152-211. DOI
4. Xiong, W.; Olson, G. B. Cybermaterials: materials by design and accelerated insertion of materials. *npj. Comput. Mater.* **2016**, *2*, 15009. DOI
5. Harbour, L.; Giudicelli, G.; Lindsay, A. D.; et al. 4.0 MOOSE: enabling massively parallel multiphysics simulation. *SoftwareX* **2025**, *31*, 102264. DOI
6. Roters, F.; Eisenlohr, P.; Kords, C.; Tjahjanto, D.; Diehl, M.; Raabe, D. DAMASK: the Düsseldorf Advanced MAterial Simulation Kit for studying crystal plasticity using an FE based or a spectral numerical solver. *Procedia. IUTAM.* **2012**, *3*, 3-10. DOI
7. Pryor, R. W. Multiphysics modeling using COMSOL?: a first principles approach. Jones & Bartlett Publishers; 2011. https://books.google.dm/books?id=Ztzs5rkytQcC&printsec=copyright#v=onepage&q&f=false. (accessed 2026-11-05).
8. Tegeler, M.; Shchyglo, O.; Kamachali, R. D.; Monas, A.; Steinbach, I.; Sutmann, G. Parallel multiphase field simulations with OpenPhase. *Comput. Phys. Commun.* **2017**, *215*, 173-87. DOI
9. Scheffler, M.; Aeschlimann, M.; Albrecht, M.; et al. FAIR data enabling new horizons for materials research. *Nature* **2022**, *604*, 635-42. DOI
10. Walsh, A. Open computational materials science. *Nat. Mater.* **2024**, *23*, 16-7. DOI PubMed
11. Ghiringhelli, L. M.; Baldauf, C.; Bereau, T.; et al. Shared metadata for data-centric materials science. *Sci. Data.* **2023**, *10*, 626. DOI PubMed PMC
12. Lucarini, S.; Martínez-Pañeda, E. UMAT4COMSOL: an Abaqus user material (UMAT) subroutine wrapper for COMSOL. *Adv. Eng. Softw.* **2024**, *190*, 103610. DOI
13. Andersson, J. O.; Helander, T.; Höglund, L.; Shi, P.; Sundman, B. Thermo-Calc & DICTRA, computational tools for materials science. *Calphad* **2002**, *26*, 273-312. DOI
14. Cao, W.; Chen, S.; Zhang, F.; et al. PANDAT software with PanEngine, PanOptimizer and PanPrecipitation for multi-component phase diagram calculation and materials property simulation. *Calphad* **2009**, *33*, 328-42. DOI
15. Sundman, B.; Kattner, U. R.; Palumbo, M.; Fries, S. G. OpenCalphad - a free thermodynamic software. *Integr. Mater. Manuf. Innov.* **2015**, *4*, 1-15. DOI
16. Otis, R.; Liu, Z. K. pycalphad: CALPHAD-based computational thermodynamics in Python. *J. Open. Res. Sofew.* **2017**, *5*, 1. DOI
17. Obaied, A.; Tang, F.; Roslyakova, I.; Baben, M. "2 1/2th" generation Calphad databases: extrapolating heat capacities of elements and compounds to 0K. *Calphad* **2021**, *75*, 102352. DOI

18. Pelton, A. D.; Chartrand, P.; Eriksson, G. The modified quasi-chemical model: Part IV. Two-sublattice quadruplet approximation. *Metall. Mater. Trans. A.* **2001**, *32*, 1409-16. DOI
19. Sundman, B.; Miani, F.; van de Walle, A.; et al. XTDB, an XML based format for Calphad databases. *Calphad* **2025**, *90*, 102849. DOI
20. Horsch, M. T.; Toti, D.; Chiacchiera, S.; Seaton, M. A.; Goldbeck, G.; Todorov, I. T. OSMO: Ontology for simulation, modelling, and optimization. In *12th International Conference on Formal Ontology in Information Systems*, Bolzano, Italy. Sep 13-17, 2021. https://ceur-ws.org/Vol-2969/paper47-FoisShowCase.pdf. (accessed 2026-11-05).
21. Scheidgen, M.; Himanen, L.; Ladines, A. N.; et al. NOMAD: a distributed web-based platform for managing materials science research data. *J. Open. Source. Softw.* **2023**, *8*, 5388. DOI
22. Evans, M. L.; Bergsma, J.; Merkys, A.; et al. Developments and applications of the OPTIMADE API for materials discovery, design, and data exchange. *Digit. Discov.* **2024**, *3*, 1509-33. DOI
23. Bayerlein, B.; Schilling, M.; Birkholz, H.; et al. PMD Core Ontology: achieving semantic interoperability in materials science. *Mater. Design.* **2024**, *237*, 112603. DOI
24. Goldbeck, G. EMMO: an ontology based on universal materials science concepts. *Zenodo* **2025**. DOI
25. Lambrix, P.; Armiento, R.; Li, H.; Hartig, O.; Pour, M. A. N.; Li, Y. The materials design ontology. *Semant. Web.* **2024**, *15*, 481-515. DOI
26. Du, Y.; Chang, Y.; Gong, W.; et al. Thermodynamic properties of the Al–Nb–Ni system. *Intermetallics* **2003**, *11*, 995-1013. DOI
27. Fujita, N.; Ishikawa, N.; Roters, F.; Tasan, C. C.; Raabe, D. Experimental–numerical study on strain and stress partitioning in bainitic steels with martensite–austenite constituents. *Int. J. Plast.* **2018**, *104*, 39-53. DOI
28. Schäfer, B. J.; Song, X.; Sonnweber-Ribic, P.; Ul Hassan, H.; Hartmaier, A. Micromechanical modelling of the cyclic deformation behavior of martensitic SAE 4150 - a comparison of different kinematic hardening models. *Metals* **2019**, *9*, 368. DOI
29. Salama, H.; Ali, M. A.; Shchyglo, O.; Steinbach, I. Phase-field simulation framework for modeling martensite and bainite formation in steel. *Comput. Mater. Sci.* **2024**, *241*, 113033. DOI
30. Hillert, M. The compound energy formalism. *J. Alloys. Compd.* **2001**, *320*, 161-76. DOI
31. Dagdelen, J.; Dunn, A.; Lee, S.; et al. Structured information extraction from scientific text with large language models. *Nat. Commun.* **2024**, *15*, 1418. DOI PubMed PMC
32. Choi, J.; Lee, B. Accelerating materials language processing with large language models. *Commun. Mater.* **2024**, *5*, 449. DOI
33. Polak, M. P.; Morgan, D. Extracting accurate materials data from research papers with conversational language models and prompt engineering. *Nat. Commun.* **2024**, *15*, 1569. DOI PubMed PMC
34. Ji, Z.; Lee, N.; Frieske, R.; et al. Survey of hallucination in natural language generation. *ACM. Comput. Surv.* **2023**, *55*, 1-38. DOI
35. Peroni, S.; Soiland-Reyes, S.; Sefton, P.; et al. Packaging research artefacts with RO-Crate. *Data. Sci.* **2022**, *5*, 97-138. DOI
36. Leo, S.; Crusoe, M. R.; Rodríguez-Navas, L.; et al. Recording provenance of workflow runs with RO-Crate. *PLoS. One.* **2024**, *19*, e0309210. DOI PubMed PMC
37. Bauer, S.; Benner, P.; Bereau, T.; et al. Roadmap on data-centric materials science. *Modell. Simul. Mater. Sci. Eng.* **2024**, *32*, 063301. DOI
38. Barker, M.; Chue, Hong. N. P.; Katz, D. S.; et al. Introducing the FAIR principles for research software. *Sci. Data.* **2022**, *9*, 622. DOI PubMed PMC

Disclaimer/Publisher's Note: All statements, opinions, and data contained in this publication are solely those of the individual author(s) and contributor(s) and do not necessarily reflect those of OAE and/or the editor(s). OAE and/or the editor(s) disclaim any responsibility for harm to persons or property resulting from the use of any ideas, methods, instructions, or products mentioned in the content.

© The Author(s) 2026. Open Access This article is licensed under a Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, sharing, adaptation, distribution and reproduction in any medium or format, for any purpose, even commercially, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.