D2.4

First report on exascale workflow infrastructure and demonstrators

A. Maslov, M. Vandelli, R. Montecucco, C. Cavazzoni, A. Marrazzo, O. Baseggio, I. Carnimeo, P. Delugas, D. Sangalli, S. Laricchia, F. Paleari, C. Cardoso, N. Spallanzani, A. Ferretti, D. Varsano, A. Garcia, A. Böhm, E. Macke

Due date of deliverable	30/06/2025 (**month 30**)
Actual submission date	dd/mm/2025
Final version	dd/mm/2025
Lead beneficiary	LEONARDO (participant number 14)
Dissemination level	PU - Public

Document information

Project acronym	MAX
Project full title	Materials Design at the Exascale
Research Action Project type	Centres of Excellence for HPC applications
EuroHPC Grant agreement no.	101093374
Project starting/end date	01/01/2023 (month 1) / 31/12/2026 (month 48)
Website	http://www.max-centre.eu
Deliverable no.	D2.4

Authors	A. Maslov, M. Vandelli, R. Montecucco, C. Cavazzoni, A. Marrazzo, O. Baseggio, I. Carnimeo, P. Delugas, D. Sangalli, S. Laricchia, F. Paleari, C. Cardoso, N. Spallanzani, A. Ferretti, D. Varsano, A. Garcia, A. Böhm, E. Macke
To be cited as	Maslov et al. (2025): First report on exascale workflow infrastructure and demonstrators Deliverable D2.4 of the HORIZON-EUROHPC-JU-2021-COE-01 project MAX (final version as of 29/06/2025). EC grant agreement no: 101093374, LEONARDO, Leonardo SpA.

Disclaimer

Contents

Executive Summary

This deliverable reports the current status of the design and implementation of exascale-ready computational workflows developed within Work Package 2 (WP2), "Exascale workflows and extreme data for materials" of the MAX Centre of Excellence. These workflows demonstrate scalable, modular, and fully reproducible automation pipelines tailored for materials science applications. The report highlights several workflow demonstrators that leverage MaX codes such as Quantum ESPRESSO and Yambo, and related simulation tools.

One demonstrator focuses on high-throughput simulations for catalyst screening, employing QUANTUM ESPRESSO and `AiiDA`'s WorkGraph extension to study the dehydrogenation of formic acid on metal surfaces. This use case integrates surface relaxation, adsorption energy evaluation, and reaction pathway analysis via NEB calculations into a single, traceable workflow.

A second demonstrator addresses the growing role of machine learning in atomistic simulations by introducing `AiiDA-TrainsPot`, a newly developed workflow for the automated training of neural network interatomic potentials (NNIPs). Combining DFT, molecular dynamics, and active learning, `AiiDA-TrainsPot` has been validated through a collaboration with CINECA by a hero run on the full Leonardo supercomputer Booster partition, demonstrating its efficiency and stability.

The third major demonstrator is `AiiDA-hubbard`, a workflow for computing Hubbard $+U+V$ parameters from first principles using DFPT. Applied to over 100 Li-based materials, this workflow enables accurate electronic structure corrections in transition-metal compounds, supporting predictive simulations without empirical fitting. In addition, we present a snapshot of the implementation work on other substantial workflows for non-adiabatic molecular dynamics, exciton-phonon coupling, and photo-catalysis.

By combining automation, provenance tracking, and interoperability, these workflows mark a significant step toward scalable, data-centric computational science and demonstrate the scientific and industrial value of MAX software in the exascale computing landscape.

1 Introduction

The transition to exascale computing is reshaping the landscape of materials science, enabling researchers to simulate increasingly complex systems with unprecedented accuracy and throughput. To fully exploit these emerging capabilities, there is a critical need for scalable, reproducible, and interoperable computational workflows that can bridge specific scientific inquiry with modern and evolving HPC infrastructure. This deliverable presents the ongoing progress made within the WP2 of the MAX Centre of Excellence, which focuses on the development and deployment of such workflows, specifically targeting exascale infrastructures and extreme data exploitation for materials simulation.

Some of the work shown in this report builds on the `AiiDA` platform, a widely adopted open-source engine for workflow automation and data provenance. The demonstrators leverage recent developments such as AiiDA WorkGraph, which enables a flexible, modular approach to composing workflows, as well as the power of novel WorkChain strategies for the integration of neural network potential training, and the enhancement of the accuracy of DFT methods via first-principles determination of Hubbard parameters for advanced simulations. As mentioned in D2.1, `AiiDA` is a key asset in our toolchest for workflow development, together with the MaX flagship codes and meta-scheduling tools such as HyperQueue. It provides a robust foundation for a set of demonstrators that address both scientifically and industrially relevant challenges.

Three main workflow demonstrators are presented. The first is an industrial-oriented high-throughput simulation pipeline for catalytic surface reactions, specifically designed to evaluate the adsorption energies and activation barriers involved in the dehydrogenation of formic acid, a promising liquid hydrogen carrier. By integrating `AiiDA` WorkGraph with QUANTUM ESPRESSO, the workflow orchestrates large-scale surface and reaction pathway calculations while ensuring full provenance and scalability.

The second demonstrator explores the automated construction of neural network interatomic potentials (NNIPs) through a first-principles active-learning loop. This framework, named `AiiDA-TrainsPot`, leverages both QUANTUM ESPRESSO and the classical Molecular Dynamic engine, LAMMPS, to sample atomic configurations and train machine-learned potentials using MACE. It is validated through large-scale "hero runs" on the Leonardo supercomputer (CINECA), demonstrating readiness for full exascale deployment.

The third major component is the AiiDA-Hubbard workflow, which automates the calculation of Hubbard $+U+V$ parameters from first principles using DFPT. These parameters are essential for correcting self-interaction errors in DFT simulations of transition-metal and rare-earth compounds. AiiDA-Hubbard combines automation with physical accuracy and has been applied to over 100 Li-based materials, uncovering trends that are critical for battery materials design.

In addition to these demonstrators, the report includes updates on the design and implementation of other showcase workflows dealing with non-adiabatic molecular dynamics, exciton-phonon coupling, and photo-catalysis, as well and supporting infrastructure developments, such as enhancements to HyperQueue for efficient task scheduling and data transfer, and new workflow components for emerging functionals and data types.

Altogether, this deliverable provides a comprehensive snapshot of the MAX CoE's strategy and efforts toward exascale workflow readiness. By combining advanced automation, robust data management, and high-performance computing, the workflows pre-

sented here demonstrate how open-source tools and HPC integration can boost scientific discovery and support industrial innovation at the exascale frontier.

2 Industrial demonstrators of exascale workflows

In the next sections, we present a discussion of an industrial demonstrator in the context of MAX exascale workflows. The first two sections will focus on selecting the industrial case, while the subsequent ones will demonstrate how the MAX lighthouse code QUANTUM ESPRESSO, integrated with the AiiDA WorkGraph tool, can play a crucial role in designing such a workflow.

2.1 Background explanation on the industrial need for novel catalysts

The fast technology advancement together with the increase in energy demands and the growing environmental concerns has increased the interest in developing sustainable and efficient energy sources, which represent one of the most relevant challenges of the 21st century. In this context, hydrogen (H_2) stands out as a promising next-generation fuel for long-distance transportation, building heating systems, and as a powerful tool for industrial synthesis processes[1]. Indeed, H_2 utilization produces only water as a by-product without releasing any greenhouse gases while generating three times the amount of energy of ordinary diesel and six times that of coal[2]. Although these characteristics look promising, the development of the H_2 economy faces two major challenges, among others [3]:

1. **H_2 Production**: Up to now, H_2 is mainly produced from fossil fuels (grey hydrogen from gas reforming and black hydrogen from coal gasification). On the other hand, blue hydrogen produced with natural gas reforming combined with CO_2 capture represents the greenest alternative up to date. Despite the latter producing H_2 with reduced environmental impact and reasonable price there is room for a new technology to further reduce both carbon footprint and production costs.

2. **H_2 Storage**: The very low volumetric energy of H_2 increases the level of difficulty of obtaining effective storage at ambient conditions. Despite several technologies have been proposed for large-scale H_2 storage, they all present several drawbacks such as significant energy losses or high pressures / low temperatures required for operations.

Therefore, a new technology aiming to disrupt the new H_2 economy should be able to provide green and *in situ* H_2 production as well as effective storage and transportation with minimum risk and energy losses. Chemical hydrogen storage in the form of liquid organic hydrogen carriers (LOHCs) represents one of the most suitable candidate technologies. For example, formic acid (FA, HCOOH) can be employed as an H_2 carrier due to its high volumetric energy density, low toxicity, low flammability, and wide availability (global production achieves millions of tons per year). Using heterogenous catalysis, H_2 can be efficiently produced from FA at ambient conditions with relative facility. Furthermore, a carbon-neutral H_2 storage/release cycle consisting of de-hydrogenation from FA to H_2 and CO_2 and hydrogenation back to FA can be built by using appropriate catalyst.

The first FA dehydrogenation experiment was conducted in the 1930s and since then there has been a continuous improvement of the catalytic performances. Up to date, noble metals such as gold, iridium, platinum and palladium among others, represent the majority of catalytic active centres, since they provide high selectivity (FA dehydrogenation is preferred towards FA dehydration), remarkable CO-tolerate capacity, and excellent

turnover frequency (TOF). Noble metals can also be combined to form binary or ternary alloys, further increasing both the catalytic efficiency and the catalyst stability. However, the scarcity and cost of these metals combined with their large price hinders the industrial large-scale application of this technology. Therefore, research is nowadays directed towards the development of non-noble metal catalyst with similar performances. Consequently, there is need for a powerful computational tool to predict, to model and to simplify the discovery of novel non-noble metal-based heterogeneous catalyst comprising high selectivity and poisoning tolerance (large turnover number TON), and excellent TOF with low cost and wide market availability. The use of such computational methods can reduce the amount of time needed for the catalyst development, as well as the cost of chemicals and the time resources needed to perform the experiments.

2.2 Towards Formic Acid Dehydrogenations: Elementary reactions steps

Several elementary reaction steps have been identified for formic acid (HCOOH) decomposition on a given metal surfaces [4, 5]. The overall reaction can proceed via dehydrogenation (yielding $CO_2 + H_2$) or dehydration (yielding $CO + H_2O$). The dehydrogenation route is usually more desirable since CO is a catalyst poison and can be detrimental for the catalyst performances. Below we review the key steps, and their known mechanisms:

- **Formic Acid Adsorption**: HCOOH can adsorb either molecularly or dissociatively on metal surfaces. On transition metals, it is assumed to dissociate upon adsorption into a surface formate (HCOO*) and an adsorbed H atom. The molecular binding to the surface is relatively weak, whereas dissociative adsorption to HCOO* + H* is thermodynamically favored on reactive metals at high operating conditions. Possible adsorption configurations are monodentate (via one oxygen) or bidentate (via both oxygens) attachments to the surface, while a parallel orientation, relative to the surface, is generally the least favorable configuration.

- **O–H Bond Scission (Formate Formation)**: Breaking the O–H bond of HCOOH is the most common initial step, producing an adsorbed formate species (HCOO*) and a surface H atom. This O–H cleavage pathway generally have the lower activation barrier compared to C–H cleavage. The resulting formate species is usually bidentate, bound through both oxygens to metal atoms, which is its most stable configuration on metallic surfaces [6].

- **C–H Bond Scission (Carboxyl Formation)**: An alternative initial step is breaking the C–H bond of formic acid, yielding an adsorbed carboxyl (HOCO*) intermediate plus a surface H. This "carboxyl route" forms a species with the hydroxyl group still attached to the initial molecular species. The C–H bond scission is typically higher in energy (thus kinetically less favorable) than the previous O–H scission. Nevertheless, the carboxyl (COOH*) intermediate is of great interest: if it forms, it can either further dehydrogenate to CO_2 or undergo other reactions (see below).

- **Direct C–O Bond Scission (CO Pathway)**: Formic acid could, in principle, decompose by cleaving its C–O bond directly, which corresponds to a dehydration pathway producing CO and OH (or O) on the surface. However, direct C–O cleavage from molecular HCOOH is highly unfavorable, it is unlikely to dissociate di-

rectly into CO (plus H_2O) in a single step. Instead, CO formation typically occurs via subsequent decomposition of another intermediate (such as COOH* or HCOO*), rather than a one-step C–O scission of formic acid.

Regarding the reactions that the formic acid undergo, the indirect dehydration path (via intermediates) is the relevant route for CO, while direct C–O breaking of HCOOH is not a significant elementary step on most metals. After the formic acid, we now look at the formate reaction pathways on a metallic surface.

- **C–H Bond Scission in Formate ($HCOO^- \rightarrow CO_2 + H$)**: Once formate (HCOO*) is formed on the surface, the critical step to complete dehydrogenation is breaking the C–H bond. This step produces CO_2 (which typically desorbs or weakly binds) and another H atom on the surface. Formate C–H scission is generally the rate-limiting step for the H_2 + CO_2 pathway, since formate is often very stable on the surface. This step may have a significant activation barrier depending on the metallic substrate. Nonetheless this represents the key step that ultimately liberates CO_2. The two H atoms generated (one from O–H cleavage, one from this C–H cleavage) can later recombine to produce H_2.

- **C–O Bond Scission in Formate (Dehydration to CO + H_2O)**: In addition to formate's C–H breaking, a competing fate of formate could be the C–O bond scission, which leads to CO formation (a dehydration pathway). In this scenario, the carbon–oxygen bond of HCOO* breaks, leaving CO (adsorbed) and a surface O atom (or O–H group). That O will scavenge H from the surface to form OH or H_2O (ultimately producing water). This pathway is generally less favourable than the previous formate dehydrogenation. This direct conversion of HCOO* to CO is difficult unless a hydrogen or hydroxyl is available to assist in making H_2O [7]. Nevertheless, on strongly oxophilic metals (or in the absence of pathways to remove the second H), formate could potentially decompose by this route, leaving surface oxygen that might poison or oxidize the catalyst.

- **Formate Binding Modes and Isomerization (Bidentate $\leftrightarrow$ Monodentate)**: An important subtle step is the possible reorientation of the adsorbed formate. Formate on metals typically prefers a bidentate configuration (both O atoms bound to metal sites) which maximizes stability. However, for formate to decompose (C–H break to CO_2), it may need to convert to a monodentate configuration at or before the transition state. The bidentate formate is very rigid and strongly bounded to the surface, whereas the transition state for C–H scission is more "loosely" attached (higher entropy). In other words, the formate likely lifts one oxygen off the surface (becoming more like a monodentate species) as the C–H bond breaks [6, 8].

- **Hydrogen Recombination and Desorption**: The H atoms produced in the above steps (from O–H and C–H bond cleavage) diffuse and pair up on the metal surface to form H_2, which then desorbs from the surface. This last step typically find low barriers for $H^* + H^* \rightarrow H_2$ on metals, making it not not rate-limiting under reaction conditions. One potential caveat is if the surface presents high coverage of adsorbed species, then the H recombination could be hindered. This step completes the catalytic cycle by releasing the stored hydrogen as H_2 gas.

2.3 Extensions of AiiDA: Workgraph concept and its functionality

Understanding the elementary reaction steps involved in formic acid decomposition provides the theoretical basis for modelling catalytic activity across different metal surfaces. However, capturing these mechanisms at the atomic scale requires a robust and scalable simulation framework. QUANTUM ESPRESSO is particularly well-suited for this task, offering state of the art implementations for both density functional theory and reaction pathway analysis via the Nudged Elastic Band (NEB) method. All bundle with GPU acceleration for optimal performances on modern HPC architectures. Yet, the complexity and volume of calculations needed to explore adsorption sites, molecular orientations, and reaction barriers across a combinatorial space of materials call for a high level of automation. This is where AiiDA plays a crucial role. As a workflow engine with built-in data provenance and high-throughput capabilities, AiiDA enables systematic and reproducible screening of catalytic surfaces. The introduction of the WorkGraph extension further enhances this capacity by allowing to define flexible, modular workflows that efficiently orchestrate the various computational tasks involved in catalyst screening.

In fact, the energy required to initate the reaction of the (de-)hydrogenation of Formic Acid can be computed using QUANTUM ESPRESSO by combining two different kinds of calculations: the self-consistent calculation implemented in the pw application, which is used to determine the energy of a static arrangement of atoms, and the neb application [9], which finds the reaction path by using the Nudged elastic band method (NEB).

To efficiently construct and manage the large number of calculations involved in the workflow for this catalytic application, we leverage the AiiDA workflow ecosystem, an open-source Python framework designed to streamline, automate, and manage high-throughput computational workflows, particularly in materials science [10, 11]. AiiDA supports workflow automation, data provenance tracking and data management based on the FAIR principles (Findable, Accessible, Interoperable, Reusable). The open-source nature of AiiDA has pushed the creation of a rich ecosystem of plugins for interfacing with major simulation packages in materials science, including the MAX codes.

Specifically, we use the recently developed AiiDA-WorkGraph extension [12], which introduces the concept of a WorkGraph for defining workflows in a highly modular and flexible way. Unlike traditional AiiDA WorkChain classes, which require writing a new workflow outline (sequence of steps) for each protocol, WorkGraph allows users to build workflows by connecting tasks (being individual calculations or entire sub-workflows) in a graph-like fashion. This design simplifies the implementation of complex sequences of QE calculations while preserving all AiiDA's features such as automatic checkpointing and data provenance. Each task in a WorkGraph is a self-contained process (for example, a PwCalculation or even a PwBaseWorkChain) with well-defined inputs and outputs. The WorkGraph then orchestrates these tasks by passing outputs from one task as inputs to another, automatically establishing dependencies between them. This dataflow approach means that the logic of the workflow is abstracted into the connections between tasks rather than a single monolithic function, yielding a framework that is more maintainable and customizable. AiiDA WorkGraph imposes no special constraints beyond core AiiDA's requirements (inputs/outputs must be AiiDA data nodes), ensuring full compatibility with existing plugins. The end result is a highly modular and reusable workflow definition: complex simulations can be constructed from smaller building blocks, enabling to compose advanced workflows with minimal new code.

An intresting application of WorkGraph's flexibility is the integration of a NEB protocol into a QE workflow. The NEB method usually involves multiple calculations in sequence, typically one first relaxes the initial and final structures and then performs an NEB calculation to find the minimum energy path between the two structures. Using a WorkGraph, these steps can be defined as separate tasks and linked seamlessly. For instance, one could even define a `PwRelaxWorkChain` task for the initial structure and another `PwRelaxWorkChain` for the final structure and then, once those are completed, feed their relaxed `StructureData` output as inputs into a `NebCalculation` task (the QE neb.x) as the start and end structures for the reaction path. `AiiDA`-WorkGraph then seamlessly ensures that the NEB task will automatically wait for and receive the outputs of the relaxation tasks (through the connected data dependencies) without the need to explicitly pass data via context variables or write custom logic. This modular design not only makes the workflow easier to read and maintain, but also easily extensible, additional intermediate relaxation steps or phonon calculation on the Transition state TS geometry can be inserted simply by adding new tasks and defining the logical connections in the WorkGraph.

In summary, WorkGraph provides a flexible and customizable framework for orchestrating advanced QE simulations like NEB calculations, by breaking them into interchangeable components. Researchers can thus tailor complex workflows (involving many sub-calculations and conditional steps) by rearranging or extending the WorkGraph, rather than coding an entirely new WorkChain for each variation.

2.3.1 Demonstrated workflow examples (`PWBaseWorkchain`)

To illustrate the benefits of the WorkGraph, it is useful to compare it with a traditional WorkChain approach on a representative QUANTUM ESPRESSO workflow. A well-known example is the SCF + band structure workflow as implemented in the `PwBandsWorkChain` of the `AiiDA` Quantum ESPRESSO plugin. This conventional workflow (a subclass of WorkChain) executes a sequence of steps to obtain electronic band structures: it relax the crystal structure (if requested), then use SeeK-path [13] to symmetrize the cell and get a standardized primitive cell, then proceed to run an SCF calculation on that structure, and finally compute band energies along high-symmetry k-points. In the WorkChain implementation, these steps are encoded in the `spec.outline` of the `PwBandsWorkChain` class, with Python methods like `run_relax`, `run_scf`, texttt-trun_bands defining each stage and `if_` logic handling optional parts. Data passing between steps is done via the workchain context (or by storing outputs of sub-workchains and then retrieving them in subsequent steps), and the overall logic is fixed by the outline order. While this approach works soundly, it has also some inherent limitations: writing a new WorkChain for each variation demands a good amount of programming effort and yields rigid workflows, a strict logical path to be followed.

In fact, WorkChains can be inflexible for diverse applications. This approach, where each unique workflow is like building a dedicated trail, leads to duplicated efforts for processes that may share certain sub-steps. Consequently, even minor modifications, such as adding a cell optimization or relaxation, might necessitate creating entirely new WorkChains, thus increasing maintenance burden.

Using WorkGraph for the same SCF + bands scenario leads to a more generic and reusable setup. Instead of a fixed outline, one would create a WorkGraph and add the cor-

responding tasks: a `PwRelaxWorkChain` task for relaxation, a `seekpath_structure_analysis` task for symmetry analysis, a `PwRelaxWorkChain` task for the SCF, and another `PwBaseWorkChain` for the bands calculation. These tasks are then linked by connecting their inputs and outputs, letting the WorkGraph execute this network of tasks respecting the dependencies (for example, automatically ensuring the SCF runs only after the structures are ready). This means that there is no need to write explicit *"glue code"* to shuttle data between steps or to define a rigid execution order. The PwBandsWorkChain example has in fact been demonstrated to translate naturally into a WorkGraph with each step as a node in the graph, confirming that WorkGraph can achieve the same functionality with a higher level of abstraction.

Below we highlight some structural and functional differences distinguishing the WorkGraph approach from classical WorkChains:

- **Workflow Definition Paradigm**: WorkChain workflows are defined by writing a new Python class with a predetermined sequence (`spec.outline`) of steps and imperative logic. In contrast, WorkGraph workflows are defined declaratively by assembling tasks and their connections. This means less complexity as it does not require writing new classes for each protocol. The workflow is essentially a graph of processes, which can be visualized and understood at a glance in terms of dependencies.

- **Reusability and Composition**: the reusability in the WorkGraph is achieved by treating each simulation step as a pluggable component. Existing WorkChains (like `PwBaseWorkChain` or `PwRelaxWorkChain`) or calculations (`PwCalculation`, `NebCalculation`) can be directly used as single tasks in a larger scale WorkGraph. A complex workflow can thus be built by composing pre-existing pieces rather than coding new ones. In a classic WorkChain, combining different sequences often necessitates developing a new workflow from scratch: a new train line for each new destination. With WorkGraph, one can mix and match tasks for different purposes (e.g., swap a relaxation method, add a convergence check step, or run multiple analysis in parallel) without reinventing the entire workflow logic.

- **Logic Abstraction and Maintainability**: In the WorkGraph, the high-level logic (which steps to do, in what order or under what conditions) is abstracted into the graph configuration, separate from the task implementations. This separation makes the workflows easier to maintain and modify. For example, to add an additional calculation step or a conditional branch, one can insert a new task node or conditional link, rather than editing a sequence of Python methods inside a WorkChain class. Overall, WorkGraph workflows tend to be more concise and easier to adapt to new requirements, since changes in one part of the graph do not require rewriting the entire control logic.

- **Parallelism and Execution Flow**: Because WorkGraph is inherently a directed acyclic graph of tasks, independent tasks can be launched in parallel if they have no mutual dependencies. This can optimize execution of complex workflows. A WorkChain, on the other hand, typically executes steps sequentially as written in the outline, unless the developer explicitly manages parallel dispatch of subprocesses right from the beginning. WorkGraph's scheduler can handle concur-

rency and synchronization implicitly based on the graph structure, which is particularly useful for workflows that have branches (for instance, exploring multiple structures or parameters concurrently). In addition, WorkGraph retains `AiiDA`'s automated checkpointing at each task node, meaning that long simulations are robust to interruptions just as WorkChains are.

In summary, WorkGraph vs. WorkChain can be seen as a shift from a static, code-defined workflow to a dynamic, graph-defined workflow writing. For Quantum ESPRESSO users, this means that complex simulation protocols (from basic SCF loops to intricate NEB reaction pathway searches) can be implemented with greater ease. The WorkGraph extension provides a maintainable and extensible framework that encapsulates QE calculations as modular nodes, thereby simplifying the orchestration of advanced workflows without sacrificing the provenance and reliability features of `AiiDA`. This flexibility ultimately accelerates development and deployment of new simulation workflows, since researchers can focus on scientific logic and composition rather than boilerplate workflow code.

2.4 Workflow design

The following industrial use case demonstrator is designed for high-throughput computational screening of the catalytic process related to the dehydrogenation of formic acid (HCOOH) on metallic surfaces. It employs the `AiiDA` WorkGraph API as the orchestration engine, coupled with QUANTUM ESPRESSO for PW and NEB calculations. We have previously stressed how the dehydrogenation of formic acid is a key step in hydrogen storage and release technologies. Simulating this reaction pathway on various metal surfaces helps identify optimal catalysts with low energy barriers and favourable adsorption characteristics. In this demonstrator we streamline the evaluation of surface and gas-phase reference energies, adsorption energies at various adsorption sites and molecular orientations and Transition States search via Nudged Elastic Band (NEB) calculations. The overall goal is to populate a reaction landscape across different metal facets and intermediates with minimal manual intervention. The entire simulation is encapsulated in a single WorkGraph, which acts as a flexible directed acyclic graph (DAG) of interdependent tasks. Each node in the WorkGraph represents either a Quantum ESPRESSO calculation (`PwCalculation`, `NebCalculation`) or a post-processing `calcfunction` that computes derived quantities such as adsorption or activation energies. The output nodes from surface and molecule calculations are directly used as inputs for the adsorption and NEB steps. By systematically looping over different surfaces, metals, adsorption sites, and molecular configurations, the workflow generates hundreds of interconnected nodes in a structured and fully traceable manner. For each combination of metal (e.g., Cu, Ag) and crystal face (100, 110, 111), an fcc slab is constructed using the ASE software library in Python [14]. Molecules, such as formic acid, formate ion, hydrogen, and carbon dioxide, are relaxed in vacuum and used as reference states for comparisons. Adsorption structures are generated by placing molecules at various adsorption sites (*ontop*, *bridge*, *hollow*) on the surface. For linear molecules, as well as for the formic acid and formate ion multiple rotations are sampled (e.g., H_up, H_down , *para* i.e. parallel to the surface and *ortho* i.e orthogonal to the surface) using user-defined rotation functions. Each configuration becomes an independent `PwCalculation` with position constraints on

the bottom layers of the slab to simulate bulk behaviour. A `calcfunction` named $adsorption\_E$ evaluates the adsorption energy as:

$$E_{\text{ads}} = E_{\text{ads/slab}} - E_{\text{slab}} - E_{\text{mol}} \tag{1}$$

This step is parallelized across multiple inputs, for streamlining execution.

After this three classes of NEB calculations are defined: Formic Acid to intermediates, Formate isomerization, Formate decomposition and Hydrogen recombination. Each NEB uses previously computed adsorbed configurations as endpoints while a custom `neb_activation_energy CalcFunction` extracts the energy barrier from the NEB outputs. All nodes producing adsorption or activation energies are labeled using a consistent schema that includes metal, face, site, molecule, and reaction type. This labeling is critical for downstream data collection, filtering, and later visualizations. This WorkGraph-based workflow demonstrator is an example of modern, modular scientific automation. It abstracts away the complexity of manually orchestrating `PwCalculation` and `NebCalculation` simulations across combinatorial chemical spaces, while preserving transparency, traceability, and adaptability. Its current design enables rapid prototyping of catalytic systems, easy addition of new metals or pathways, and seamless integration into data-driven catalyst screening pipelines.

2.5 Preliminary results

In this section, we present some preliminary results of the industrial high-throughput demonstrator for the computational screening study designed to investigate the reaction mechanisms and energetics of formic acid dehydrogenation on metallic surfaces. The workflow was executed on the proprietary Leonardo S.p.a. HPC infrastructure, DaVinci-1. Specifically, each calculation was run on a single `gpu` node equipped with 2 AMD EPYC Rome 7402 @ 2.80 GHz CPUs (24 cores each), 512 GB of RAM and 4 NVIDIA A100 GPUs with 40 Gb of RAM each. QE v7.3.1 with GPU support was used for the calculations [15] and kenrel version of `AiiDA` is the 2.6. The test prototype of the workflow consisted of 36 `pw` and 12 `neb` individual executions.

The workflow encompasses multiple classes of reactions associated with the dehydrogenation and decomposition of formic acid, using two metal surfaces, Cu(110) and Ag(110), representative of a common metal and noble metal respectively. The goal is to evaluate adsorption energies and activation barriers across various elementary steps involved in the transformation of formic acid and its reaction intermediates. Precisely, the calculations include: adsorption studies for all molecular species on a single adsorption site (*ontop*) for each metallic surface and NEB calculations for selected reaction paths involving molecular transformations and dissociation events. The molecules considered span reactants, intermediates, and dissociation products relevant to formic acid dehydrogenation.

These molecules were adsorbed on the (110) surfaces of Cu and Ag, using a single representative adsorption site (*ontop*) to reduce complexity during this preliminary screening. The workflow accounted for various rotational configurations where applicable (e.g., formate with *H_up* and *H_down*). Charged species like formate and its dissociation products were treated using a corrected total charge input in the QE setup.

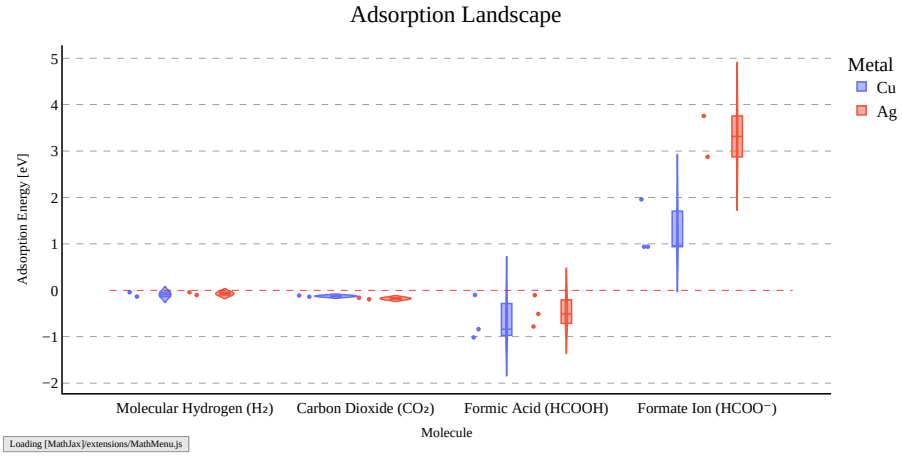

Figure 1: "Adsorption energy distributions for selected molecular species on Ag(110) and Cu(110), evaluated at the *ontop* adsorption site. Violin widths represent the spread of energies across different molecular orientations and configurations.

Figure 1 presents the computed adsorption energies for the molecular species investigated in this study, adsorbed on the *ontop* site of the (110) surfaces of silver and copper. These results provide an initial overview of how each adsorbate interacts with the metal surface under a fixed site and orientation, serving as a reference point for evaluating trends in binding strength and surface reactivity. In a full production workflow, this dataset would be significantly expanded to include multiple metals, various crystallographic orientations (such as (100) and (111) facets), and all relevant adsorption sites (including bridge and hollow positions fro example). Such a comprehensive screening would allow for a more complete mapping of the adsorption landscape, ultimately supporting more accurate predictions of catalytic activity and selectivity across different material candidates.

To illustrate the next level of detail enabled, Figure2 shows the adsorption energies resolved by adsorption site for each molecular species on both Ag(110) and Cu(110) surfaces. By moving beyond a single-site analysis, this visualization highlights how adsorption strength varies not only between metals but also across different local binding environments, for now only the *ontop* site, but in a later step also for *bridge*, and *hollow* sites. The site-specific resolution is essential for understanding surface reactivity and identifying the most catalytically active configurations, particularly in systems where binding preferences are strongly site-dependent.

In addition to adsorption studies, the workflow included several key NEB calculations aimed at estimating the activation energies of relevant reaction steps. Each NEB transition was defined between an initial and final geometry, both obtained from the preceding adsorption relaxation steps. The following classes of elementary reactions were considered:

- **Formic acid dissociation**, involving both O–H and C–H bond cleavage pathways.
- **Formate transformations**, including C–H bond scission, C–O bond scission, and isomerization between different binding modes (e.g., bidentate to monodentate).

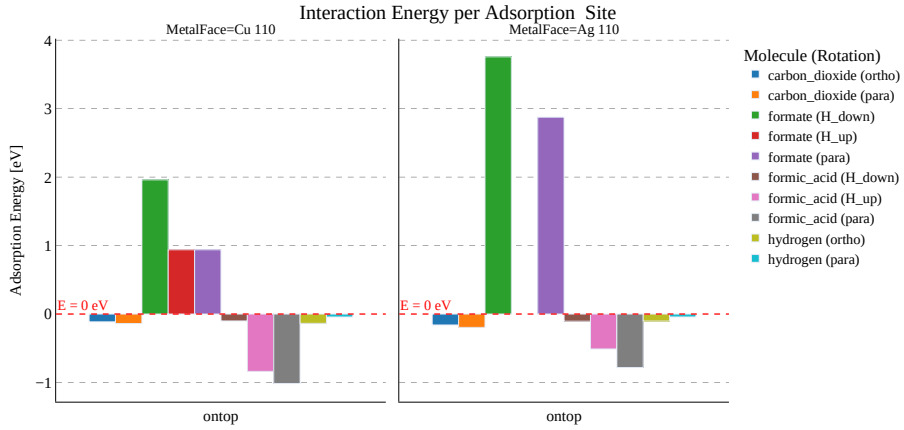

Figure 2: Adsorption energy histograms for selected molecular species on Ag(110) and Cu(110), resolved by adsorption site *ontop*. This representation highlights site-specific binding trends across both metal surfaces.

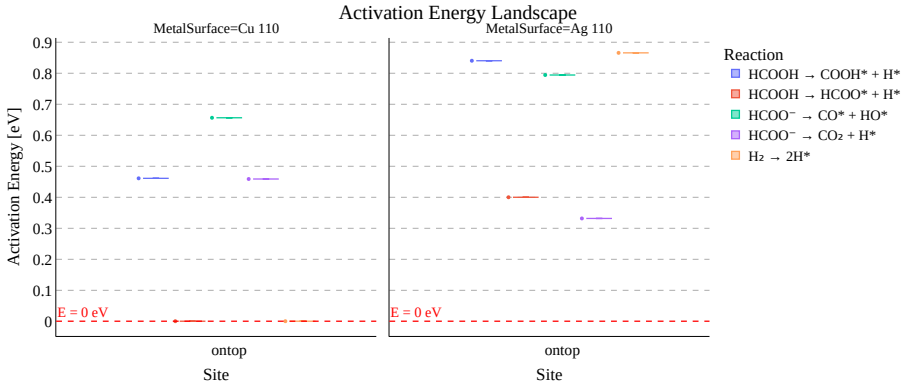

Figure 3: Activation energy distributions for selected NEB pathways on Ag(110) and Cu(110), grouped by adsorbate. Violin widths reflect the spread in barrier heights across different sites and configurations.

- **Hydrogen recombination and desorption**, modeling the formation and release of H_2 from adsorbed H atoms.

For each metal-surface pair (Cu(110) and Ag(110)), NEB transitions were set up when the required endpoint geometries were available from the adsorption workflow. In total, the NEB component comprised approximately 12 distinct reaction pathways, each discretized using 4 intermediate images, resulting in around 48 individual self-consistent field (SCF) relaxations handled via `pw.x`. It is worth noting that one of the explored pathways—the formate binding-mode isomerization—produced an unphy sically high energy barrier exceeding 20 eV. This suggests either an unfavorable transition path or convergence issues in the interpolation and was therefore excluded from the summary plots in Figure 3 and Figure 4.

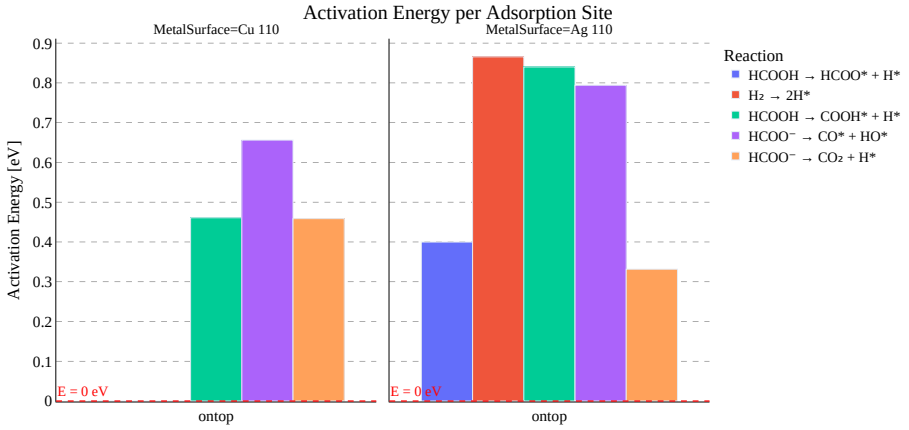

Figure 4: Site-resolved activation energy barriers for NEB transitions on Ag(110) and Cu(110).

2.6 Estimate of the computational effort of the workflow

High-throughput simulations, while offering systematic insight into complex phenomena, come with a substantial computational cost. Here, we present a quantitative estimate of the computational effort required to execute the workflow described previously, focusing on time-consuming components such as PW relaxations and NEB (nudged elastic band) calculations.

The evaluation provides both an overview of resource distribution across different calculation types and an estimate of per-task cost, which is crucial for planning scalability, reproducibility, and further implementation that will be needed for the exascale capacity. The total wall time and average time per calculation were extracted from the `.out` and `.time` files generated by QUANTUM ESPRESSO. A custom parser module was developed to collect the data directly from the AiiDA provenance graph using the node labels and UUIDs associated with each completed `PwCalculation` and `NebCalculation`. The focus was towards two primary calculation classes:

- `PwCalculation` for surface relaxations, molecule-in-vacuum relaxations, and adsorbate relaxations;
- `NebCalculation` for computing minimum energy paths and activation energies for reaction mechanisms.

Lighter `@calcfunction` nodes such as the arithmetic calculation of Adsorption Energy were excluded from this effort estimate, as they require negligible compute time in comparison.

Figure5 presents the total elapsed compute time for all the calculation type within the workflow. While adsorption relaxations account for a significant portion of the wall time, NEB calculations represent a substantial computational cost despite appearing later in the timeline. This is because NEB tasks are inherently dependent on the completion of preceding relaxation steps, as they require relaxed initial and final geometries to define the reaction path. In the WorkGraph based execution, this dependency is encoded as a directed dataflow, meaning that NEB tasks can only be launched once all required

upstream relaxations are completed. As a result, their runtime contribution is not only shaped by their iterative nature and multiple-image setup but also by their position within the workflow's execution order. In contrast, surface and gas-phase molecule relaxations are relatively lightweight and run early in the workflow, without having the burden to wait for previous calculation thus having limited impact on the overall runtime profile.

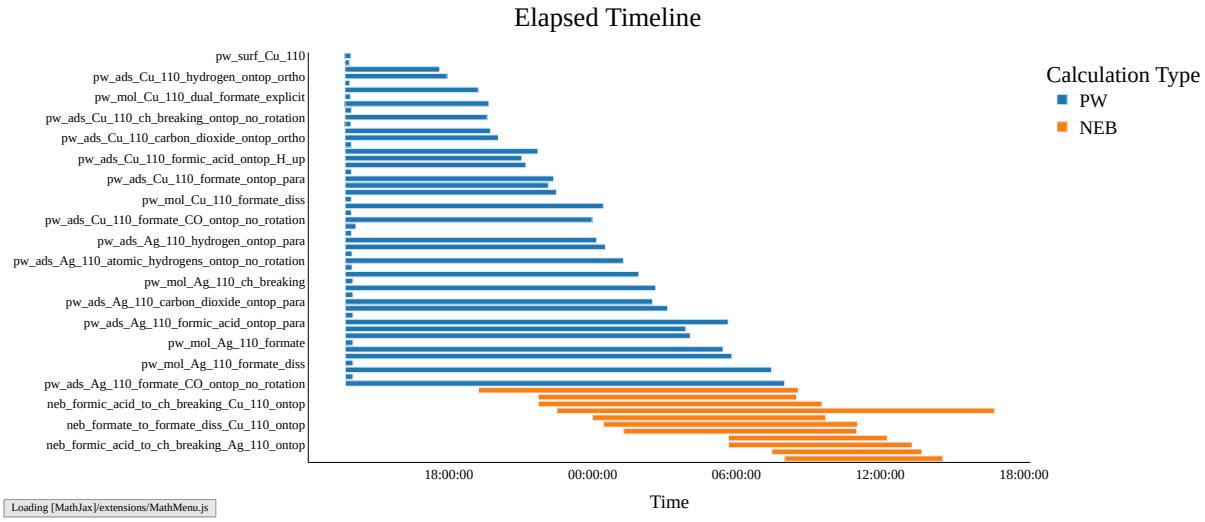

Figure 5: Total elapsed compute time per calculation type of the workflow.

The second figure (Figure6) presents the aggregated wall time per calculation, grouped by task type. Although NEB tasks appear less dominant in total wall time compared to standard `PwCalculation` relaxations (111.4 h vs. 311.1 h), this is primarily due to the simplified setup used in this pilot phase, where each NEB transition was discretized using only 4 images. In a full production workflow, a higher number of images (typically 9 to 11) would be employed to improve resolution along the reaction coordinate, leading to a significantly greater computational burden per NEB transition. Therefore, while NEB currently accounts for about one-third of the total compute time, its relative cost is expected to grow substantially in high-resolution runs.

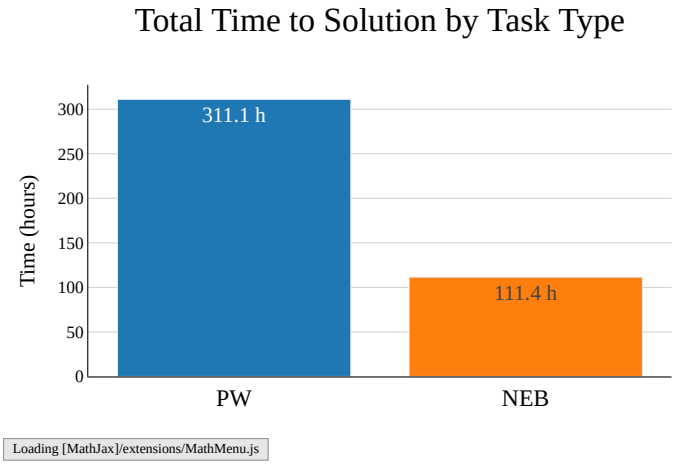

Figure 6: Wall time per calculation type.

2.7 Summary remarks

The workflow presented in this study, although currently demonstrated for two metals (Cu and Ag) and a single surface orientation (110), has been designed with scalability and modularity at its core. Its architecture allows straightforward extension to a much broader chemical and structural space, including multiple metals, alloy compositions, crystallographic facets (e.g., 100, 111), and a range of adsorption sites and molecular orientations.

The data collected in this pilot phase provide a valuable benchmark for estimating the computational effort required in full production-scale campaigns. A realistic extension to 10 metals and 3 surface facets, for instance, would increase the job count by at least an order of magnitude, demanding fine-grained parallelization strategies and efficient resource scheduling. Further expansion to alloy systems in real catalytic materials, would easily push the workload into a regime that could saturate HPC with limited capacity.

Scaling this workflow to such complexity will therefore require access to larger compute infrastructures, ideally exascale platforms, where the full benefits of modular, data-driven orchestration can be realized. Beyond its technical robustness, this workflow demonstrates a concrete path toward systematic, reproducible catalyst screening with direct industrial relevance. Its deployment on upcoming exascale facilities could mark a critical step forward in accelerating the discovery and optimization of sustainable catalytic materials at scale.

3 Other workflow demonstrators: design and implementations

3.1 Automated learning of neural-network interatomic potentials

Neural network interatomic potentials (NNIPs) have significantly advanced atomistic simulations by enabling molecular dynamics (MD) with near *ab initio* accuracy at a fraction of the computational cost and with enhanced scalability. However, the construction of high-fidelity NNIPs remains a non-trivial endeavor, requiring deep expertise in both machine learning and electronic structure methods. In addition, the construction of a good training dataset requires running thousands of *ab initio* simulations, those become hundreds of millions if universal NNIPs—i.e., capable of interpolating across a broad range of elements, compositions, and structures—are sought for [16].

SISSA, in collaboration with the ICSC National Research Centre for High Performance Computing, Big Data and Quantum Computing and University of Trieste, has been developing `AiiDA-TrainsPot`, an exascale `AiiDA`workflow to facilitate the efficient generation of accurate NNIPs; a corresponding scientific publication is in preparation [17]. Thanks to the support of the MaX CoE and collaborations with one of its partner CINECA, SISSA has been able to test earlier versions of `AiiDA-TrainsPot` on the entire Leonardo ($\approx$ 12k NVIDIA A100 GPUs) supercomputer, hence validating the approach, identifying bottlenecks and the corresponding solutions to produce an exascale-ready `AiiDA` WorkChain. `AiiDA-TrainsPot` tightly integrates density functional theory (DFT) calculations with classical MD to systematically sample the potential energy surface via a diverse set of atomic configurations, including random distortions, strain, surface and interfacial structures, neutral point defects, atomic clusters, and MD trajectories spanning a range of thermodynamic conditions. Central to the approach is an active learning protocol that dynamically guides data acquisition by calibrating committee model disagreement against ground-truth DFT errors in an on-the-fly fashion, thereby ensuring both accuracy and data efficiency.

3.1.1 The `AiiDA-TrainsPot` WorkChain

The logical and software structure of AiiDA-TrainsPot is shown in Fig. 7. In a typical use case, users provide a handful of input structures—from one up to tens—that are augmented to the order of thousands through structural manipulations: other supported use cases include providing a large pre-existing dataset of structures and/or DFT calculations as well as a pre-trained NNIP (e.g., a foundational model) to be fine-tuned by `AiiDA-TrainsPot` for the systems of interest. In the most general scenario, the first stage takes care of calculating a thousand of structures with *ab initio* methods (typically, but not necessarily, DFT) for training the first generation of NNIPs. The second stage leverages molecular dynamics (MD) to accurately sample the potential energy surface around equilibrium. A number of structures are sampled from MD trajectories and calculated *ab initio*, then training a new generation of NNIPs. Structures to label are chosen based on committee disagreement, that is the spread in the predictions of a committee of NNIPs initialized with different seeds but trained on the same dataset. Notably, at each iteration we calibrate this statistical uncertainty on the true error, i.e., the deviation from DFT predictions: this yields a more reliable error estimate to be deployed even for systems too large to be simulated *ab initio*.

The AiiDA-TrainsPot software architecture is very modular and designed to maximize both end-to-end automation and the selective execution of individual components for different tasks. The entire workflow is orchestrated by the top-level TrainsPotWorkChain, which coordinates five specialized sub-workchains corresponding to the major stages of the active learning cycle: DatasetAugmentationWorkChain for sampling diverse structural configurations, AbInitioLabelingWorkChain for *ab initio* labeling, TrainingWorkChain for training the committee of NNIPs, ExplorationWorkChain for exploring the potential energy surface using MD and EvaluationWorkChain for committee evaluation. AbInitioLabelingWorkChain leverages the AiiDA-QUANTUM ESPRESSO package and in particular the PwBaseWorkChain, adopting as default the Standard Solid-State Pseudopotentials (SSSP) protocol [18] for the choice of pseudopotentials, plane-wave cutoffs and k-point density. The ExplorationWorkChain is based on AiiDA-LAMMPS, that was slightly improved to support calculations with NNIPs. The training of NNIPs with MACE [19] required developing from scratch a dedicated AiiDA Plugin (MaceTrainCalculation) and WorkChain (MaceTrainWorkChain), which were embedded in the higher-level TrainingWorkChain. Thanks to the AiiDA's plugin interface, AiiDA-TrainsPot leverages all the available AiiDA turn-key solutions that automate and streamline calculations with QUANTUM ESPRESSO and LAMMPS, which essentially prevented almost entirely duplication of efforts and maximizes interoperability. In addition, thanks to the hierarchical and modular structure of our design, support for other *ab initio* or NNIP engines could be done with limited efforts.

In spirit of facilitating non-experts in using AiiDA-TrainsPot, a simple 10-step tutorial (see Fig. 8) has been prepared and will be made available at the time of publication and software release.

3.1.2 Hero runs and exascale training of interatomic potentials

The scalability and performance of AiiDA-TrainsPot has been tested with two hero runs on the full Leonardo supercomputer (Booster partition), in partnership with CINECA. During the first hero run (see left panel of Fig. 9), we demonstrated the excellent scalability of AiiDA, which efficiently utilizes the machine's full capacity and dynamically adapted when more resources became available. The submitted jobs included an ensemble of DFT calculations for 3500 materials through the AiiDA WorkChains for QUANTUM ESPRESSO, with a diverse set of input parameters and number of atoms, as well as a AiiDA-TrainsPot WorkChain for the training of a NNIP with $\approx$ 1800 supercell DFT calculations for hundreds of atoms in the unit cell. In addition to validating the scalability of AiiDA and of its automated workflows, these data are contributing to CINECA's efforts to develop performance models for QUANTUM ESPRESSO.

During the second hero run (see right panel of Fig. 9), we launched a single AiiDA-TrainsPot WorkChain on the entire Leonardo supercomputer (about 3k (12k) nodes (GPUs)). We note that all calculations, namely DFT calculations with QUANTUM ESPRESSO, MD simulations with LAMMPS and NNIPs training with MACE, are run on GPUs with GPU-accelerated software. The WorkChain was able to exploit the full machine during the first phase of the *ab initio* labelling, where thousands of QUANTUM ESPRESSO calculations were launched, each running on 4 nodes and 16 GPUs with 16 MPI tasks and 8 OMP threads per task. After few tens of minutes, AiiDA could exploit about half of the cluster, probably due to the limited computational resource of the workstation hosting the AiiDA

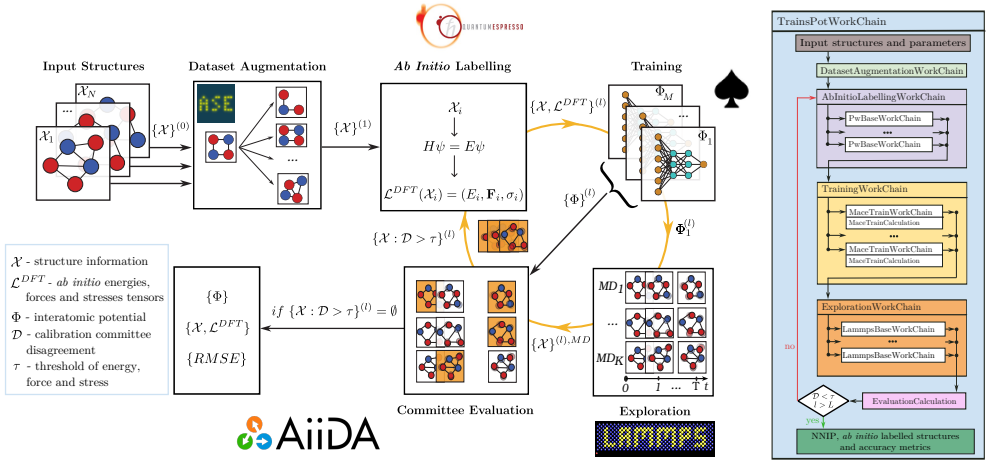

Figure 7: Schematic representation of the `AiiDA-TrainsPot WorkChain` for the automated training of neural-network interatomic potentials. After initialization, the `DatasetAugmentationWorkChain` generates additional structures and the workflow enters an active learning loop. Within each iteration, the `AbInitioLabellingWorkChain` labels the newly generated configurations by performing DFT calculations with QUANTUM ESPRESSO via the `PwBaseWorkChain`. The `TrainingWorkChain` is then invoked to train NNIPs using MACE through the `MaceTrainWorkChain`. Subsequently, the `ExplorationWorkChain` performs MD simulations using LAMMPS (`LammpsBaseWorkChain`) to generate new configurations for further refinement. The `EvaluationCalculation` assesses model performance based on committee disagreement to determine whether additional active learning iterations are needed to further improve the NNIPs.

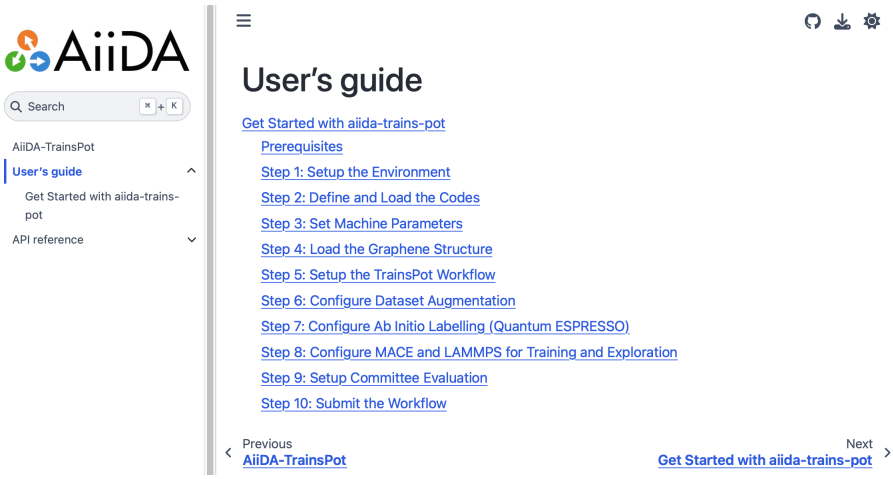

Figure 8: Homepage of the `AiiDA-TrainsPot` web user guide, that includes a complete tutorial from installation to deployement for scientific applications.

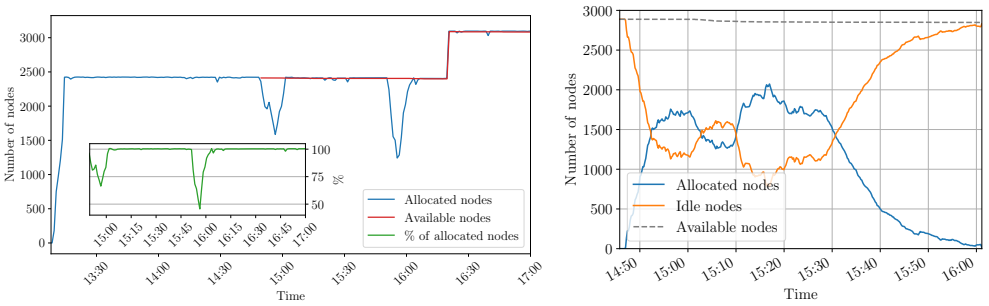

Figure 9: Hero runs with AiiDAon the Leonardo supercomputer at CINECA, Booster partition. Left panel: First run, submitted jobs includes DFT AiiDA QUANTUM ESPRESSO calculations for 3500 materials through the pw.x WorkChains, with a diverse set of input parameters and number of atoms, plus an AiiDA-TrainsPot WorkChain for the training a NNIP with ≈ 1800 DFT calculations for supercells containing hundreds of atoms. Right panel: Second hero run, single AiiDA-TrainsPot WorkChain launching ≈ 1100 DFT calculations in the first iteration of the active learning loop.

daemon and its workers. Due to the limited time window during which we had access to the full cluster, the training and exploration parts were not run, but we noted that the day after AiiDA required a few hours to fetch all the simulations from the cluster and prepare the input for training NNIP potentials with MACE. These results and observations suggest that for exascale deployment also the server running AiiDA should be highly-performant in terms of compute capacity, memory (RAM) and connection bandwidth: many of these conditions could be met by installing an AiiDA instance within the network of the supercomputing centres, possibly through virtual machines or docker containers hosted on dedicated servers or internal cloud services. Finally, the long tail in the number of occupied nodes shown in Fig. 9 (right-hand panel) is due to the wide range of computing time required by the different DFT simulations: the AiiDA daemon has to wait all simulations (here about 1100) to be finished before proceeding with the next step. We think this bottleneck will not be present in relevant exascale applications, where the number of calculations will be much higher than the number of nodes (from tens of thousands to hundreds of millions for training universal interatomic potentials).

In the second hero run, we also experienced bottlenecks related to storing and loading all the structures, energy, forces and stress tensors that were produced by QUANTUM ESPRESSO or LAMMPS. The solution was to introduce a dedicated AiiDA datatype with the desired properties and adopting the HDF5 file format, as discussed next.

3.1.3 PESData: A custom AiiDA datatype for potential energy surfaces

AiiDA-TrainsPot deals with heterogeneous collections of atomic structures, these collections often include a mix of structures of different compositions, cell shapes, and number of atoms. Moreover, each structure may be associated with different kinds of metadata, such as DFT-calculated energies, forces, and stresses, as well as predictions from a committee of interatomic potentials. The standard TrajectoryData type in AiiDA assumes that all structures in a dataset have the same atom count, which is unsuitable for this more flexible use case. To address this and other limitations, we introduced the PESData class.

`PESData` is a subclass of `aiida.orm.Data`, specifically designed to store and manage a list of `ase.Atoms` structures—potentially heterogeneous—along with their associated physical properties. It serializes the dataset as an HDF5 file which is persisted in the `AiiDA` repository, enabling scalable storage and efficient access even for large datasets. Here we list some of the key features of the `PESData` class:

- Flexible structure support: Each item in a `PESData` dataset can have a different number of atoms and structure type. This contrasts with `TrajectoryData`, which is limited to a given system.
- Metadata: For structures with an attached `SinglePointDFTCalculator`, the class automatically extracts and stores energy, forces and stress tensor in a structured HDF5 format. The design supports future expansion to include predictions by an ensemble of machine-learned potentials, as well as other relevant properties.
- Iterator Interface: The class implements `__iter__()` and `__next__()` to allow Pythonic iteration over stored structures, as well as a memory-efficient generator method `iter_items()` for large datasets. This avoids loading at once the full dataset in memory, avoiding out-of-memory issues and improving overall performance.
- Efficient I/O with HDF5: Data is written and read from HDF5 files using the h5py library, and serialized to the AiiDA repository using a UUID-tied key for uniqueness. The PESData class uses an HDF5 group format, where each structure is stored in a group labelled `item_i`, and each group contains datasets for cell vectors, atomic symbols, positions, and optionally, energy, forces, and stress. Attributes such as the number of labelled/unlabelled frames and the chemical species set are stored in the node's `AiiDA` attributes for efficient summary access.
- Concatenation support: Two PESData nodes can be merged using the + or += operator, yielding a new node that combines the structures and metadata from both inputs.
- Compatibility with ASE: The `set_ase()` method supports input from lists of `ase.Atoms` objects and can recognize whether a structure has DFT metadata embedded in its calculator.

3.2 Hubbard parameters from first-principles with automated and reproducible workflows

Most of the challenges to achieving accurate predictions of material properties with DFT are buried in the exchange-correlation (xc) functional, whose exact analytical form is unknown and must therefore be approximated. Among the numerous xc functionals proposed, the local-density approximation (LDA) and the (semilocal) generalized-gradient approximation (GGA) have become the most popular choices, mainly for efficiency reasons. However, despite their successful application to a large variety of systems, LDA and GGA functionals have proven much less adequate for the treatment of transition-metal (TM) and rare-earth (RE) compounds. These issues originate from electron self-interaction errors (SIEs), which particularly plague the description of partially occupied and localized d and f states.

So-called Hubbard corrections to DFT constitute a conceptually simple and inexpensive way to mitigate SIEs by means of one or several corrective terms that modify the DFT total energy and the potential of the localized states to which they are applied (e.g., $3d$ shells). The strength of the correction is gauged by the numerical values of the associated Hubbard parameters U and V, which increase the localization of states centred on atoms (U) and centred on bonds between states (V), respectively.[1] In the past, the numerical values of these Hubbard parameters were often determined empirically, usually by adjusting them such that specific observables (like the band gap) were brought to match experimental data points. However, reliance on empirical fitting procedures severely restricts the predictive capabilities of Hubbard-corrected DFT, as they cannot be applied to undiscovered compounds. Moreover, it is important to note that Hubbard parameters are not transferable between different systems and computational setups. Therefore, instead of using empirical Hubbard parameters, the values of U and V should always be calculated from first principles. A popular approach for this purpose is based on density functional perturbation theory (DFPT) which is physically equivalent to but computationally more efficient than the older, well-established linear response constrained DFT (LR-cDFT) approach. Hubbard U and V parameters determined from DFPT often lead to significant improvements in the predictions of electronic, thermodynamic and structural properties, with only a marginal increase in computational cost compared to uncorrected DFT [20, 21, 22, 23, 24, 25, 26, 27]. Unfortunately, obtaining a converged set of Hubbard parameters from DFPT can require a significant number of individual calculations including geometry optimizations, SCF steps with and without occupation smearing, and several perturbative DFPT calculations.

This complexity severely limits the potential of DFT+U + V for exascale applications, e.g., with regard to the automated discovery of novel Li-containing compounds for battery applications.

3.2.1 The `AiiDA-hubbard` workflow

To diminish the burden in applying DFT+U + V with non-empirical Hubbard parameters on the exascale, we have implemented `AiiDA-hubbard` [28], a python-based plugin for `AiiDA` that provides the first optimized and fully automated workflow to perform structurally self-consistent calculations of Hubbard U and V parameters. Currently, `AiidDA-hubbard` interfaces the PW.X and HP.X codes [29] of the QUANTUM ESPRESSO distribution, used for SCF and DFPT calculations, respectively. The workflow, shown in Fig. 10, contains several key building blocks. The main self-consistent Hubbard workflow and its subprocesses are implemented as `AiiDA` *workchains*, powering the automated handling of calculations and errors as well as ensuring full reproducibility. These workchains are represented by grey rectangles. Data nodes, containing inputs and outputs of the workflows and calculations, are depicted by green rounded boxes. For clarity, not the entire nested list of inputs is shown in the figure, but only the mandatory input data classes needed to run the workflow. The light grey box (Fig. 10a) contains the outline of the `SelfConsistentHubbardWorkChain`, the main workflow of the package, which carries out the self-consistent calculation of the Hubbard parameters. Its child processes are the `PwBaseWorkChain` and

[1] Note that several other local corrective terms, such as Hund's J are also known and used in practise

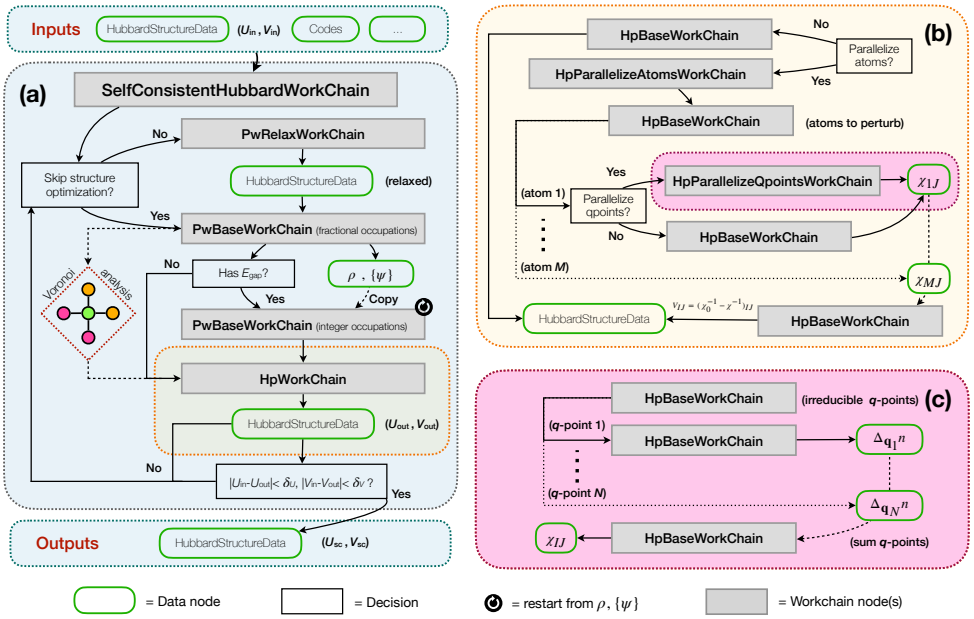

Figure 10: **Schematic illustration of the `aiida-hubbard` plugin that automates the self-consistent calculation of the Hubbard U and V parameters.** (a) The main `SelfConsistentHubbardWorkChain` workflow which automates the self-consistent calculation of U and V parameters. It iterates (optionally) structural optimizations via the `PwRelaxWorkChain`, ground-state calculations via the `PwBaseWorkChain`, and the DFPT calculations of Hubbard parameters via the `HpWorkChain`. In particular, the latter can be used to fully exploit the parallel capabilities of the HP code, i.e. by (optionally) first parallelizing the calculations over inequivalent Hubbard atoms to perturb, using the `HpParallelAtomsWorkChain` (panel (b)), and then (optionally) parallelizing over irreducible monochromatic perturbations ($\mathbf{q}$ points) via the `HpParallelQpointsWorkChain` (panel (c)). These nested calls are visualized by the different colored boxes.

the `PwRelaxWorkChain`, which are specialized workchains that run the PW code as part of the `aiida-quantumespresso` plugin [30], and the `HpWorkChain` managing the parallel capabilities of the HP code. The orange and pink boxes (Fig. 10b and c) zoom in on the fine-grained parallelization facilitated by the DFPT framework and the HP code. The main input and output of the workflow is a `HubbardStructureData` object, a new data type created to store information on the Hubbard functional together with the atomistic structure.

For improved user-friendliness, `AiiDA-hubbard` features a `get_builder_from_protocol` method for each of the workchains [30, 31]. This method automatically populates the inputs, leaving the user left with the task of providing only three indispensable pieces of information: (i) an instance of `HubbardStructureData`, (ii)`AiiDA` code instances containing information on how to run the PW and HP codes [10, 11], and (iii) a string defining in a general fashion the accuracy of the calculation called `protocol` (“fast”, “balanced”, or “stringent”) [32]. After calling the `get_builder_from_protocol` method, the user receives a pre-populated set of inputs, which can then be checked and

modified before being used for the execution of the workflow.

3.2.2 `HubbardStructureData`: a joint description of Hubbard interactions and atomistic structures

`HubbardStructureData` unifies the description of Hubbard corrections and the respective atomistic structures in one class and provides user-friendly auxiliary utilities that facilitate the initialization and handling of Hubbard-related data. This is achieved by combining the structural information, inherited from the `StructureData` class already available in AiiDA, with a new `Hubbard` class composed of three key components: the mathematical formulation of the correction to be applied (*flavour*), the Hubbard projectors (the mathematical representation of the subspace to which the corrections are applied), and the numerical values of the interaction parameters, stored as a list of instances of `HubbardParameters`. `HubbardParameters` is an extra class defining a single Hubbard interaction that contains its type (U, V, J etc.), the indices and manifolds (e.g. $3d$, $2p$, etc.) of the atom(s) involved, as well as the value of the respective parameter expressed in energy units (eV). To allow for the description of Hubbard V interactions between two distinct atoms (intersite interactions), `HubbardParameters` additionally stores a second atomic index and a second target manifold. The first atomic index is always referenced within the unit cell, while every second atomic index is augmented by a translation vector $\mathbf{t}$ pointing to the atom's corresponding periodic image. The high level of abstraction makes `HubbardStructureData` a FAIR and code-agnostic data type that is expected to be useful beyond the boundaries of this first automated workflow for Hubbard parameters.

3.2.3 Application of `AiiDA-hubbard` to $>$ 100 Li-containing solids

To showcase the scalability and robustness of the `AiiDA-hubbard` workflow, we carried out calculations of Hubbard U and V parameters across a diverse set of 115 known crystalline solids containing Li as well as Fe or Mn with unit cells of 32 atoms or fewer. This choice of elements was based on the pivotal role of lithium-ion batteries for modern technology, particularly in powering a wide range of devices. All but ten of the submitted `SelfConsistentHubbardWorkChain` processes finished successfully, often owing to the effective automated recovery of computational errors that occurred in about 6% of all the DFT+U+V and DFPT calculations submitted. In addition to demonstrating the robustness of the workflow, these calculations revealed important physical trends. For example, Figure 11, which shows the range of self-consistent Hubbard U parameters determined by the successful workchains, clearly indicates a general dependence of U on the oxidation state (OS) of the TM elements. Furthermore, additional analysis (not presented in this report for conciseness) indicate correlations between the numerical values of U and V on the one hand and bond lengths, coordination numbers and ligand species on the other hand. These insights, suggestive of a complex dependence of U and V on both structural *and* electronic degrees of freedom, are extremely valuable for future machine learning models aimed at the prediction of Hubbard parameters. Finally, the `AiiDA-hubbard` workflow is expected to be particularly useful for modelling large systems (e.g., with defects) or for calculating observables whose evaluation potentially requires numerous independent calculations, such as vibrational properties.

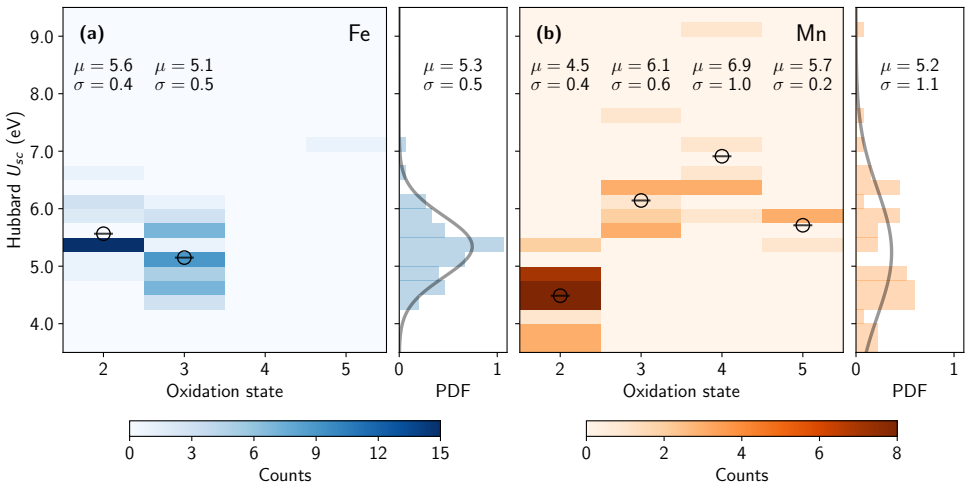

Figure 11: **Distributions of self-consistent Hubbard U_{sc} parameters among 105 Li-bearing materials. (a,b)** Values of onsite Hubbard U_{sc} parameters as a function of the oxidation states of Fe and Mn. The side panels show the data probability distribution along with the fitted Gaussian distributions (gray lines) of U_{sc} across the explored oxidation states and report the mean values (μ) and standard deviations (σ) in units of eV.

Both of these kinds of applications benefit from the enhanced description of the electronic structure due to Hubbard U and V corrections, which incur minimal additional computational cost compared to the underlying functionals. This prospect is further supported by the extendable, modular and code-agnostic nature of the package, and even more so by the constantly growing universe of `AiiDA` plugins, including those outlined in the present report.

3.3 Non-adiabatic molecular dynamics for excited states

The first step for the development of a NAMD worklow has been accomplished, with the implementation of the TD-DFPT transition energy gradients. This has been a major development in the QUANTUM ESPRESSO suite, that required an extensive refactoring of the `turbo_davidson` and `phonon` codes. Gradients have been derived from full TD-DFPT equations, retaining the possibility to recast the Tamm-Dankoff approximation as an optional feature.

The Z-vector scheme, originally formulated for CIS gradient [33] and already well exploited in other codes [34, 35] has been adapted to the QUANTUM ESPRESSO plane-wave formalism. A solver for the Z-vector has been developed, based on the conjugate gradient approach to solve one set of Sternheimer (CPHF) equations instead of 3N. Advantageously, the Z-vector solver has been developed starting from the one already implemented in the PHonon code at Gamma (`phcg.x`). This required some major code refactoring to move the relevant routines in modules shared between TD-DFPT and PHonon. The reusability of the code has been improved and duplications have been kept minimal.

The reliability of the gradient has been tested by direct comparison between numerical and analytical force components of the first excited state of ten small molecules (Ethene, Acetylene, Benzene, Water, Formaldehyde, Acetaldehyde, Acetone, Ethanol,

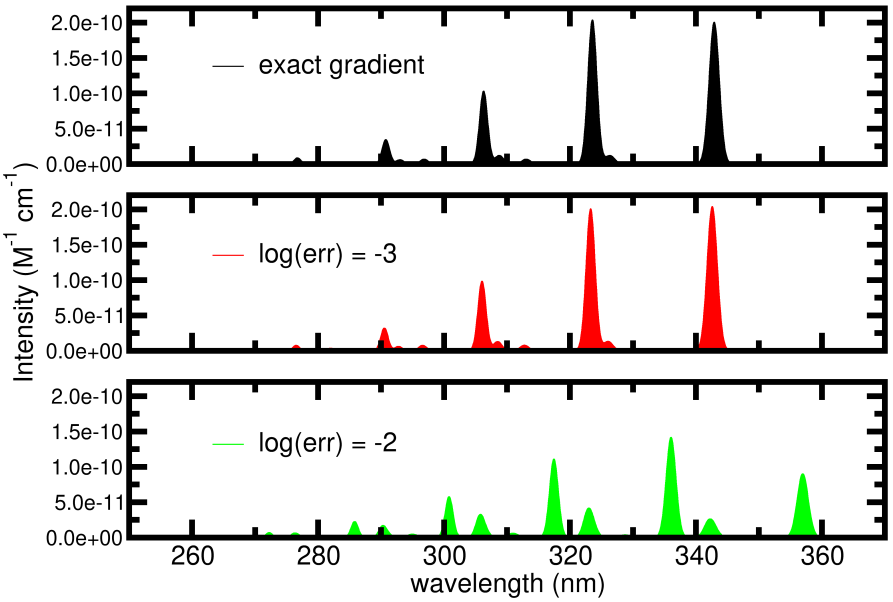

Figure 12: **Vibronic spectrum of gas phase Formaldehyde**, for the S0→S1 transition (HOMO→LUMO, $n \rightarrow \pi^*$), computed with the QUANTUM ESPRESSO suite at PBE/ONCV/60Ry level of theory, using the vertical gradient approach implemented in `FCClasses3` code [36, 37]. Middle (red) and bottom (green) panels report the change in the spectrum due to the addition of random errors to the TD-DFPT gradient of the order of 10^{-3} and 10^{-2} Ry/Bohr, respectively.

Acetamide, Acetic Acid), finding average deviation of the order of 10^{-4} Ry/Bohr. Such deviations are considered acceptable for all purposes, as shown by the sensitivity test shown in Fig. 12.

In this Figure, the vibronic spectra of the Formaldehyde molecule has been computed using the vertical gradient approach, with the program `FCClasses3` code [36, 37], and it is noted that the excited state properties start to change significantly only when errors on the gradient reach 10^{-2} Ry/Bohr. Development are still in progress to finalize the TD-DFPT gradient for hybrid functionals, and the GPU offload of the code. There is also an additional workflow in preparation, to perform excited state geometry optimizations.

3.4 XC-functional sub-workflow

This workflow constitutes a demonstrative framework for executing total energy and forces calculations for density-corrected functionals. These functionals promise to yield chemical accuracy results by combining various standard calculations [38, 39]. As such, they are susceptible to being made more efficient when embedded in a concurrent workflow. The main example of such a type of method is the DC-SCAN functional for wa-

ter [40]. For each atomic configuration, the assumed correct electronic density is computed by performing a pure Hartree-Fock calculation, and the total energy is then evaluated using the SCAN functional with the electronic charge and kinetic energy densities yielded by the Hartree-Fock calculation. The total energy computed in this way is not variational, meaning that the energy depends linearly on virtual or small variations of the HF density, and such linear dependency contributes to the forces. Using Lagrange multipliers, it is possible to compute a general response function, the so-called Z-vector, and with it to cast this contribution in a Hellmann-Feynmann form that avoids the computation of density and orbital responses for all virtual displacements [34].

Implementing the Z-vector computation is still underway. In the meantime, our proof of concept workflow utilises a simplified approach computing such variations with finite differences for a multiplier λ. In the following we very briefly sketch the method and its rationale. The DC-SCAN functional can be seen as the $\lambda = 0$ point of the parametrised functional:

$$E^{\lambda}_{AUX}[\hat{\rho}^0_{HF}] = E_{HF}[\hat{\rho}^0_{HF}] + \lambda \cdot \left[E^{SCAN}_{XC}[\hat{\rho}^0_{HF}] - E^{HF}_{X}[\hat{\rho}^0_{HF}]\right] \ . \tag{2}$$

Derivatives with respect to an external parameter τ, such as an atomic coordinate, are then:

$$\left.\frac{\partial E^{\lambda}_{AUX}}{\partial \tau}\right|_{\hat{\rho}=\hat{\rho}^{\lambda}_0} = \frac{\partial E_{HF}}{\partial \tau} + \lambda \cdot \left.\frac{\partial \left[E^{SCAN}_{XC} - E^{HF}_{X}\right]}{\partial \tau}\right|_{\hat{\rho}=\hat{\rho}^{\lambda}_0} . \tag{3}$$

The derivatives of these auxiliary forces evaluated at $\lambda = 0$ are the additional term that has to be added to the Hartree-Fock forces to compute the gradients of the DC-SCAN functional. At present, the workflow thus works by

- computing for each atomic configuration total energies and forces for

$$\lambda \in \{-0.1, -0.05, 0.00, 0.05, 0.01\}$$

- evaluating DC-SCAN total energy by feeding the charge density and orbitals from the $\lambda = 0$ Hartree-Fock calculation to a Harris-Foulkes calculation with the SCAN functional. This step will be eventually integrated directly as a final step of the $\lambda = 0$ calculation.
- computing the non-variational part of the forces by applying a least square fitting to the forces of the auxiliary functional as λ varies in our five-value range.
- next atomic positions are computed with the DC-SCAN energy and forces.

The current preparatory framework is written in Python based on ASE [14] and can perform molecular dynamics and BFGS structural minimisation.

3.5 Exciton-phonon coupling workflows

The implementation of exciton-phonon (exc-ph) couplings involves a complex workflow combining electron-phonon (el-ph) matrix elements and excitonic calculations, using `ph.x` (utility of the QUANTUM ESPRESSO suite) and YAMBO, respectively. During this reporting period, we have further progressed with the strategy outlined in D2.2. In

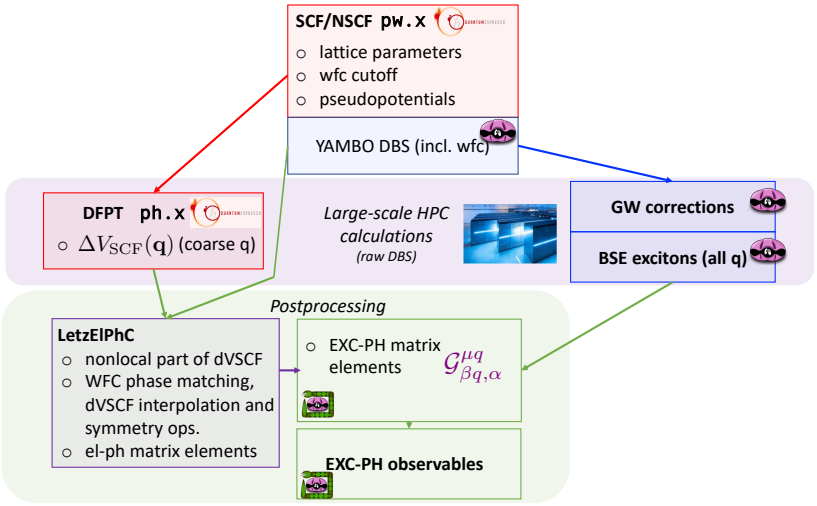

Figure 13: **Scheme of the exciton-phonon workflow**. Starting from a common DFT ground state and band structure calculation in QUANTUM ESPRESSO, the workflow branches into two computationally intensive sections: a phonon part (calculation of self-consistent potential variation followed by efficient electron-phonon matrix elements) and an exciton part (GW-BSE YAMBO scheme). Finally, the calculation of the exciton-phonon observables is a postprocessing dealt with by `yambopy`.

particular, the code `LetzElphC`, a third party external utility, has now been included as a tool in the YAMBO suite, with a dedicated repository under the YAMBO organization [41], with the main usage of handling and evaluating the el-ph matrix elements to be used in YAMBO workflows.

Indeed, `LetzElphC` has been also adapted to produce HDF5 databases in the YAMBO format, instead of requiring a conversion step which was previously done by `yambopy` [42]. Moreover, we further extended the capabilities of the `LetzElphC` code to include $\mathbf{q}$-space interpolation of the electron-phonon matrix elements. This removes, e.g., the bottleneck represented by phonon calculations for 2D systems, speeding up the exciton-phonon workflow.

Further post-processing functions have been added in `yambopy` including classes for the calculation of luminescence and lifetimes. Production runs of the workflow have also been performed to test it on realistic applications, including large 3D systems. The simulations have been successfully concluded on Leonardo. We are presently analysing the data. A scheme of the present state of the workflow, which is ready for production, is provided in Fig. 13.

A similar direction has been taken for the Ehrenfest dynamics scheme in the real-time module of the YAMBO code. Indeed, by using electron-phonon matrix elements produced by `LetzElphC`, a simplified scheme can be derived. Such implementation will be also used as a benchmark for the full fashion Ehrenfest dynamics scheme based on QUANTUM ESPRESSO pseudopotential libraries, as detailed in D2.2.

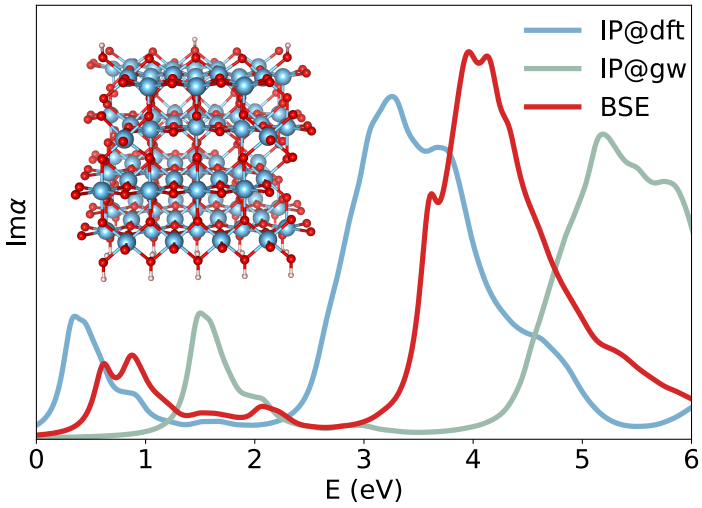

Figure 14: **Numerical output of the BSE runs in the photo-catalysis workflow**. The optical properties computed at the independent particle (IP), on top of both KS-DFT and GW results, and at Bethe Salpeter equation (BSE) level, have been obtained by the photo-catalysis workflow described in Sec. 3.6 running on a HPC partition consisting of 88 and 36 nodes (each equipped with 4 A100 Nvidia GPUs) for the GW and BSE calculations, respectively.

3.6 Large scale BSE runs for photo-catalysis workflows

This section presents a detailed computational workflow aimed at characterizing polaron photo-excitations in reduced rutile TiO_2. The motivation for analysing the optical fingerprints of polarons lies in the fact that, in reduced TiO_2, electronic transitions arising from polaronic excess electrons in localized band-gap states (BGS) are known to influence both photoabsorption and the photo-catalytic activity of TiO_2 within the visible spectrum. Density functional theory (DFT) has been widely used to investigate the electronic excited states involved in these $d \rightarrow d$ excitations [43, 44] and the BGS energy relative to the conduction band minimum. However, the accuracy of Kohn-Sham DFT is inherently limited by its theoretical framework, and, in order to address optical properties, one typically needs to move beyond DFT to TD-DFT or many-body perturbation theory (MBPT) methods. Within this work we focus on a MBPT approach.

The excess electron states induced by the donor defects may exhibit a degree of charge delocalization, being shared by several Ti atoms, but preferentially occupy specific Ti-3d orbitals, thus formally creating *small polarons* on Ti^{3+} sites [45, 46, 47, 48]. In TiO_2 rutile, small polarons are therefore expected to undergo phonon-assisted, anisotropic, incoherent adiabatic hopping [49, 50]. The adiabatic hopping mechanism of small polarons is the rationale for exploring several polaronic configurations, each of which may give rise to distinct optical fingerprints. To systematically investigate the large number of inequivalent polaronic configurations available within a reduced supercell, we developed a computational workflow combining (i) a Hubbard-corrected DFT (DFT+U) constrained structure relaxation using QUANTUM ESPRESSO (QE) – to localize the small polaron on a chosen Ti site – and (ii) a many-body approach based on

G_0W_0 and the solution of the Bethe-Salpeter Equation (BSE), offering a powerful first-principles framework to characterize optical excitations from BGS, as observed in 2PPE experiments on hydroxylated rutile TiO_2(110) surfaces [43, 44, 51] and optical absorption measurements on bulk rutile $Ti_{1-x}Nb_xO_2$ [52]. Such a workflow consists of the following steps:

- To force the localization of the excess electrons on specific Ti sites in the supercell, each geometry optimization has been performed using QUANTUM ESPRESSO within DFT+U and with a two-steps procedure following a modified version of the prescriptions reported in Refs. [53, 54, 55]: (i) a relaxation performed by artificially imposing a large value of U (e.g. 10.0 eV) at a chosen Ti site to facilitate the localization of the excess electron, followed by (ii) a second relaxation using the proper U value for all Ti sites, including the polaronic sites. In the latter relaxation the small polaron typically remains self-trapped.

- A further DFT calculation without Hubbard correction is required as starting point for the subsequent many-body scheme. Generally, we do not use the ground-state DFT+U calculation as starting point as the Hubbard correction (i) narrows bandwidths of excited t_{2g} and e_g d states and (ii) introduces a fictitious self-consistency in G_0W_0 – due to the non-local nature of the Hubbard potential – which implies an additional localization effect of the small polaron not related to structural distortions. At this step, attention must be paid with the shallow BGS, which can give rise to a fictitious metallic or semimetallic character, detrimental for the simulation of optical properties.

- Compute the G_0W_0 quasiparticle (QP) band structure using YAMBO. As generally big supercells are required to correctly describe small polarons by removing fictitious interactions with their image, a huge number of occupied and unoccupied bands are required in the evaluation of the optical spectra within the BSE scheme and in the transitions space. In order to optimize the workflow, the calculation of the electronic self-energy has been split into several runs, with the QP energies merged back in one single database using `yambopy`.

- When simulating a slab, achieving fully converged G_0W_0 calculations may be challenging because of (i) the large amount of vacuum required to minimize spurious interactions of the surface with its image, and of (ii) the difficulties in correctly modelling the screening effects at the surface. A fast and reliable convergence with respect to the sampling of the Brillouin zone (BZ) can be achieved by using the recently developed W-av method available in YAMBO [56].

- Finally, the BSE framework implemented in YAMBO was used to compute the optical spectra. The diagonalization of large BSE kernels – many high-energy transitions could be required to study the photoexcitation mechanisms involving BGS – was performed by interfacing a developmental version of YAMBO with the NVIDIA multi-process cuSOLVERMp library, a high-performance, distributed-memory, GPU-accelerated library that provides tools for the solution of dense linear systems and eigenvalue problems making use of ScaLAPACK-like C functionalities. Alternatively, YAMBO is also interfaced with a recent version of the

GPU-ported SLEPc libraries. This introduces a significant speed up in the diagonalization of large matrices, a further step in bridging the gap between pre-exascale and exascale simulations. For a more detailed description of the linear algebra interfaces available in YAMBO, have a look at Deliverable D1.4.

- Using a BSE spectrum projection tool onto defect states – justified by the negligible coupling between defect-states and valence-band transitions – distinct classes of configurations, with polaronic excitations exhibiting similar spectral fingerprints, can be identified. Interestingly, the polarons behave as isolated quasi-particles, with no optical coupling between them, and their response is influenced by the local chemical environment.

An example of the numerical results obtained by deploying the above workflow on medium/large partition of an GPU-accelerated machine is given in Fig. 14. Specifically, we have exploited a partition made of 88 and 36 nodes (each equipped with 4 A100 Nvidia GPUs) for the GW and BSE calculations, respectively.

4 Workflow tools

Deliverable D2.1 discussed our philosophy regarding tools for workflow design and implementation, based on the combination of basic components rather than a monolithic framework. We have already mentioned above the role that `AiiDA` and its extension WorkGraph have played in the development of the demonstrators. Here we report on enhancements to our metascheduler HyperQueue.

4.1 HyperQueue: Direct Data Transfers

We provide here details of a new experimental feature in HyperQueue (HQ): direct data transfers between tasks. This functionality is designed to optimize workflows characterized by numerous tasks producing small, ephemeral data objects intended for immediate consumption by subsequent tasks. By bypassing traditional distributed file systems for these transient data exchanges, this feature aims to mitigate I/O bottlenecks and improve overall pipeline efficiency.

4.1.1 Introduction and Motivation

Workflows involving a large number of computational tasks often generate intermediate data that serve solely as input for subsequent tasks. These data objects are typically small and ephemeral, meaning they have no intrinsic value beyond the immediate pipeline and do not require persistent storage. When such workflows rely on distributed file systems (DFS) for these exchanges, the high frequency of read/write operations for many small files can lead to significant performance degradation, as DFS are often not optimized for such access patterns.

The direct data transfer feature in HyperQueue addresses this challenge by providing a dedicated mechanism for managing and relaying these ephemeral data objects directly between tasks, potentially avoiding DFS overhead entirely.

4.1.2 Scope and Use Cases

This feature is specifically targeted at scenarios where:

- **Numerous tasks are involved:** The benefits are most pronounced in large-scale computations.
- **Data is ephemeral:** The produced data is temporary and primarily serves for inter-task communication. If data loss would necessitate significant re-computation or if data requires long-term storage, this feature is not the primary solution.
- **Data objects are relatively small:** While not a strict limitation, the design prioritizes efficient handling of smaller data chunks where DFS overhead is most impactful.

Conversely, this feature is *not* intended for:

- **Persistent storage of critical data:** If data integrity and long-term availability are paramount.

- **Very large data objects:** For which DFS with parallel I/O capabilities might be more suitable.
- **Workflows with few tasks:** Where the overhead of DFS interaction is likely negligible.

In such cases, utilizing a distributed file system remains the recommended approach.

4.1.3 Distinction from Existing *"Streaming"* Functionality

HyperQueue already implements a feature known as *"streaming"*, which focuses on reducing the number of files on a distributed file system by capturing `stdout` and `stderr` streams from processes. While both features aim to alleviate DFS load, they serve distinct purposes:

- **Streaming:** Primarily for capturing logging, debugging information, and progress updates typically written to standard output/error streams.
- **Direct Data Transfers:** Designed for exchanging primary computational data objects between tasks.

These two features are orthogonal and can be used concurrently.

4.1.4 Implementation Overview

The direct data transfer mechanism operates as follows:

- **Data Object Production:** A task can produce one or more uniquely identifiable data objects during its execution.
- **Data Dependencies:** Tasks can declare dependencies on data objects produced by other tasks. A consumer task will only proceed once its required data objects are available.
- **Data Management:** HyperQueue manages the lifecycle of these data objects, including their temporary storage and eventual cleanup once they are no longer needed by any dependent tasks.
- **Peer-to-Peer Transfers:** HQ facilitates efficient peer-to-peer (P2P) data transfers between worker nodes when necessary. If a consuming task is scheduled on the same node that produced the data, network communication is entirely avoided, leading to significant performance gains.
- **Scheduler Awareness:** The HyperQueue scheduler is aware of data dependencies and data locality. It endeavors to co-locate producer and consumer tasks on the same node or otherwise minimize data transfer distances and costs.

Data Object Identification: Each data object is uniquely identified by a combination of the producing task's ID and an output ID (a user-defined identifier for a specific output of that task).

Data Dependencies and Task Dependencies: HyperQueue has long supported task dependencies, where a task's execution is contingent upon the completion of one or more preceding tasks. *Note:* In the current implementation of direct data transfers, a data dependency implies a task dependency. That is, if task A depends on a data object produced by task B, task A must also wait for task B to complete. Future versions aim to provide more flexibility by potentially relaxing this strict coupling, allowing data to be consumed as it is produced, even before the producer task fully completes (streaming data objects).

4.1.5 Usage Example (Bash and TOML)

The following example illustrates how to define tasks that produce and consume data using bash scripts and a TOML configuration file.

Producing Task (`producing_task.sh`): The script performs a computation and then registers the output file with HQ.

```
#!/bin/bash
# Perform computation and generate an output file
./do_some_computation > out.data

# Upload the generated file 'out.data' to HQ,
# associating it with output ID 12
# The $HQ variable points to the path of the
# currently running HyperQueue instance.
$HQ put 12 out.data
```

Note: It is recommended that `out.data` is created in a task-specific temporary directory managed by HQ, ideally configured to use a local ramdisk (e.g., `/dev/shmem` or `/tmp` on a `tmpfs` filesystem) to maximize I/O performance and avoid shared filesystem contention. HQ offers management such task temporary directories

Consuming Task (`consuming_task.sh`): The script retrieves a data object from HQ and then uses it for its computation.

```
#!/bin/bash
# Retrieve input 0 (as defined in TOML) from HQ
# and save it as 'input.data'
$HQ get 0 input.data # The '0' refers to the index
                     # in the data_deps array

# Run computation on the retrieved data
./run_some_computation input.data
```

TOML Job Definition: This TOML snippet defines two tasks and establishes a data dependency between them.

```
# Enable the data transfer layer
data_layer = true

[[task]]
id = 1 # Producer task
```

```
command = ["./producing_task.sh"]

[[task]]
id = 2 # Consumer task
command = ["./consuming_task.sh"]

# Define data dependencies for task 2
[[task.data_deps]]
# This task's input (implicitly indexed as 0 here)
# depends on the data object produced by task with id=1
# and associated with output_id=12.
task_id = 1
output_id = 12
```

In this configuration, task `id = 2` will wait for task `id = 1` to produce the data object identified by `output_id = 12`. HQ will then make this data object available to task `id = 2` as its first input (index 0).

4.1.6 Current Status and Future Directions

The direct data transfer feature is currently in an **active testing and development phase** and is not yet available in any official HyperQueue release. We anticipate it will be particularly beneficial for complex, data-intensive workflows, such as those encountered in the MaX Centre of Excellence pipelines.

Potential areas for future development include:

- **Expanded API:** Introducing `C` and `Python` APIs for more programmatic and granular manipulation of data objects.
- **Performance Optimizations:** Further enhancements to the efficiency of data transfer protocols and scheduling strategies.
- **Flexible Data Dependencies:** Refining the definition of data dependencies, possibly decoupling them from strict task completion to allow for streaming of data objects.
- **Advanced Data Management:** Exploring features like data object versioning or checksumming for enhanced robustness.

4.1.7 Summary remarks

Direct data transfers represent a significant enhancement to HyperQueue's capabilities, offering a promising solution for optimizing the performance of workflows that heavily rely on the exchange of small, ephemeral data between tasks. By minimizing reliance on traditional distributed file systems for such transfers, this feature has the potential to substantially reduce I/O bottlenecks and improve the scalability of scientific computations.

5 Conclusions

This report has presented a set of advanced, exascale-ready computational workflows developed within Work Package 2 of the MAX Centre of Excellence. These workflows showcase the integration of automation, data provenance, and high-performance computing for cutting-edge applications in materials science. We have highlighted demonstrators covering industrial catalysis, machine-learned interatomic potentials, and first-principles corrections for strongly correlated systems. Other exascale-class workflows on non-adiabatic dynamics, exciton-phonon coupling, and photo-catalysis, are in advanced stages of development.

The demonstrated `AiiDA`WorkGraph Quantum ESPRESSO pipeline for formic acid dehydrogenation exemplifies how modular, data-centric workflows can be used to systematically explore reaction mechanisms and adsorption behaviour across diverse material surfaces. Its high-throughput design, coupled with full traceability and extensibility, lays the groundwork for scalable catalyst screening in both academic and industrial contexts.

The `AiiDA-TrainsPot` workflow for training neural network potentials highlights the power of active learning approaches in atomistic modelling. Its successful deployment on the full Leonardo supercomputer demonstrates also the scalability and maturity of the infrastructure, paving the way for future exascale campaigns in potential energy surface exploration and accelerated molecular simulations.

The AiiDA-Hubbard framework represents a major step forward in making first-principles Hubbard parameter calculations routine, reproducible, and accessible. Its application to over 100 Li-based compounds illustrates both the robustness of the approach and its relevance for electronic-structure modelling in energy materials. The YAMBO workflows for exciton-phonon coupling and photocatalysis, performing GW and BSE calculations at scale, demonstrate how large partitions of modern, GPU-accelerated, HPC machines can be exploited to directly address the simulation of scientific properties of interest.

Collectively, these efforts demonstrate the versatility and readiness of the MAX software ecosystem to support scientific discovery and industrial innovation at the exascale. The workflows presented here are not only scientifically robust and technically scalable, but also designed with interoperability and reusability to foster further development.

Looking ahead, these frameworks will serve as a foundation for further developments, including the integration of more complex models, the use of AI-driven decision-making in workflow design, and the extension of automation to post-processing and data analytics. As exascale computing becomes a reality, such workflow infrastructures will be essential to fully harness its potential, turning compute power directly into scientific insight and technological advancement.

References

[1] Wen, H. *et al.* Heterogeneous catalysis in production and utilization of formic acid for renewable energy. *Small* **20**, 2305405 (2024).

[2] Solakidou, M. *et al.* Cost efficiency analysis of h2 production from formic acid by molecular catalysts. *Energies* **16** (2023).

[3] Wang, X. *et al.* Recent progress in hydrogen production from formic acid decomposition. *International Journal of Hydrogen Energy* **43**, 7055–7071 (2018).

[4] Herron, J. A., Scaranto, J., Ferrin, P., Li, S. & Mavrikakis, M. Trends in formic acid decomposition on model transition metal surfaces: A density functional theory study. *ACS Catalysis* **4**, 4434–4445 (2014).

[5] Yoo, J. S., Abild-Pedersen, F., Nørskov, J. K. & Studt, F. Theoretical analysis of transition-metal catalysts for formic acid decomposition. *ACS Catalysis* **4**, 1226–1233 (2014).

[6] Creutzburg, M. *et al.* Heterogeneous adsorption and local ordering of formate on a magnetite surface. *The Journal of Physical Chemistry Letters* **12**, 3847–3852 (2021).

[7] Silbaugh, T. L., Karp, E. M. & Campbell, C. T. Energetics of formic acid conversion to adsorbed formates on pt(111) by transient calorimetry. *Journal of the American Chemical Society* **136**, 3964–3971 (2014).

[8] Ruehl, G., Harman, S. E., Gluth, O. M., LaVoy, D. H. & Campbell, C. T. Energetics of adsorbed formate and formic acid on cu(111) by calorimetry. *ACS Catalysis* **12**, 10950–10960 (2022).

[9] Quantum ESPRESSO Developers. *PWneb User's Guide (v. 7.2)* (2022). URL https://www.quantum-espresso.org/Doc/neb_user_guide/.

[10] Huber, S. P. *et al.* Aiida 1.0, a scalable computational infrastructure for automated reproducible workflows and data provenance. *Scientific Data* **7**, 300 (2020).

[11] Uhrin, M., Huber, S. P., Yu, J., Marzari, N. & Pizzi, G. Workflows in aiida: Engineering a high-throughput, event-based engine for robust and modular computational workflows. *Computational Materials Science* **187**, 110086 (2021).

[12] AiiDA Team. *AiiDA WorkGraph Documentation* (2025). URL https://aiida-workgraph.readthedocs.io/en/latest/autogen/quick_start.html.

[13] Hinuma, Y., Pizzi, G., Kumagai, Y., Oba, F. & Tanaka, I. Band structure diagram paths based on crystallography. *Computational Materials Science* **128**, 140–184 (2017).

[14] Larsen, A. H. *et al.* The atomic simulation environment—a python library for working with atoms. *Journal of Physics: Condensed Matter* **29**, 273002 (2017).

[15] Giannozzi, P. *et al.* Quantum espresso toward the exascale. *The Journal of Chemical Physics* **152**, 154105 (2020).

[16] MaterialsProject. Mathbench discovery (2025). URL https://matbench-discovery.materialsproject.org/.

[17] Manko, N., Bidoggi, D., Peressi, M. & Marrazzo, A. Automated learning of neural-network interatomic potentials. *in preparation* (2025).

[18] Prandini, G., Marrazzo, A., Castelli, I. E., Mounet, N. & Marzari, N. Precision and efficiency in solid-state pseudopotential calculations. *npj Computational Materials* **4**, 72 (2018).

[19] Batatia, I., Kovacs, D. P., Simm, G. N. C., Ortner, C. & Csanyi, G. Mace: Higher order equivariant message passing neural networks for fast and accurate molecular force fields. *Advances in Neural Information Processing Systems* (2022).

[20] Mahajan, R., Kashyap, A. & Timrov, I. Pivotal role of intersite hubbard interactions in fe-doped alpha-mno2. *The Journal of Physical Chemistry C* **126**, 14353–14365 (2022).

[21] Gebreyesus, G., Bastonero, L., Kotiuga, M., Marzari, N. & Timrov, I. Understanding the role of hubbard corrections in the rhombohedral phase of batio3. *Physical Review B* **108**, 235171 (2023).

[22] Haddadi, F., Linscott, E., Timrov, I., Marzari, N. & Gibertini, M. On-site and intersite hubbard corrections in magnetic monolayers: The case of feps3 and cri3. *Physical Review Materials* **8**, 014007 (2024).

[23] Timrov, I., Kotiuga, M. & Marzari, N. Unraveling the effects of inter-site hubbard interactions in spinel li-ion cathode materials. *Physical Chemistry Chemical Physics* **25**, 9061–9072 (2023).

[24] Bonfà, P. *et al.* Magnetostriction-driven muon localization in an antiferromagnetic oxide. *Physical Review Letters* **132**, 046701 (2024).

[25] Chang, B. K. *et al.* First-principles electron-phonon interactions and polarons in the parent cuprate la_2cuo_4. *arXiv* (2024).

[26] Binci, L., Marzari, N. & Timrov, I. Magnons from time-dependent density-functional perturbation theory and the noncollinear hubbard formulation. *arXiv* (2024).

[27] Malica, C. & Marzari, N. Teaching oxidation states to neural networks. *arXiv* (2024).

[28] Bastonero, Lorenzo *et al.* First-principles hubbard parameters with automated and reproducible workflows (2025).

[29] Timrov, I., Marzari, N. & Cococcioni, M. Hp – a code for the calculation of hubbard parameters using density-functional perturbation theory. *Computer Physics Communications* **279**, 108455 (2022).

[30] Huber, S. P. *et al.* Common workflows for computing material properties using different quantum engines. *npj Computational Materials* **7**, 136 (2021).

[31] Bastonero, L. & Marzari, N. Automated all-functionals infrared and Raman spectra. *npj Computational Materials* **10**, 1–12 (2024).

[32] Nascimento, G. d. M. *et al.* Accurate and efficient protocols for high-throughput first-principles materials simulations (2025).

[33] Handy, N. C. & Schaefer III, H. F. On the evaluation of analytic energy derivatives for correlated wave functions. *J. Chem. Phys.* **81**, 5031–5033 (1984).

[34] Hutter, J. Excited state nuclear forces from the tamm–dancoff approximation to time-dependent density functional theory within the plane wave basis set framework. *J. Chem. Phys.* **118**, 3928 (2003).

[35] Jin, Y., Wen-zhe Yu, V., Govoni, M., Xu, A. C. & Galli, G. Excited state properties of point defects in semiconductors and insulators investigated with time-dependent density functional theory. *J. Chem. Theory Comput.* **19**, 8689–8705 (2023).

[36] Cerezo, J. & Santoro, F. Fcclasses3, vibrationally-resolved spectra simulated at the edge of the harmonic approximation. *J. Comput. Chem.* **44**, 626–643 (2022).

[37] Santoro, F. & Cerezo, J. Fcclasses3 , a code for vibronic calculations, version 3.0.4, 2025. URL http://www.iccom.cnr.it/en/fcclasses.

[38] Vuckovic, S., Song, S., Kozlowski, J., Sim, E. & Burke, K. Density functional analysis: The theory of density-corrected dft. *Journal of Chemical Theory and Computation* **15**, 6636–6646 (2019).

[39] Song, S., Vuckovic, S., Sim, E. & Burke, K. Density-corrected dft explained: Questions and answers. *Journal of Chemical Theory and Computation* **18**, 817–827 (2022).

[40] Dasgupta, S., Lambros, E., Perdew, J. & Paesani, F. Elevating density functional theory to chemical accuracy for water simulations through a density-corrected many-body formalism. *Nature Communications* **12**, 6359 (2021).

[41] Letzelphc: A complementary c code for yambo to compute electron-phonon coupling matrix elements with full crystal symmetries. URL https://github.com/yambo-code/LetzElPhC.

[42] Paleari, F. *et al.* Yambopy (2025). URL https://doi.org/10.5281/zenodo.15012962.

[43] Wang, Z. *et al.* Localized excitation of ti3+ ions in the photoabsorption and photocatalytic activity of reduced rutile tio2. *Journal of the American Chemical Society* **137**, 9146–9152 (2015).

[44] Tanner, A. J. *et al.* Photoexcitation of bulk polarons in rutile $\mathrm{Tio_2}$. *Phys. Rev. B* **103**, L121402 (2021).

[45] Kowalski, P. M., Camellone, M. F., Nair, N. N., Meyer, B. & Marx, D. Charge localization dynamics induced by oxygen vacancies on the $\mathrm{tio_2}(110)$ surface. *Phys. Rev. Lett.* **105**, 146405 (2010).

[46] Setvin, M. *et al.* Direct view at excess electrons in $\mathrm{tio_2}$ rutile and anatase. *Phys. Rev. Lett.* **113**, 086402 (2014).

[47] Krüger, P. *et al.* Defect states at the $\mathrm{tio_2}(110)$ surface probed by resonant photoelectron diffraction. *Phys. Rev. Lett.* **100**, 055501 (2008).

[48] Minato, T. *et al.* The electronic structure of oxygen atom vacancy and hydroxyl impurity defects on titanium dioxide (110) surface. *The Journal of Chemical Physics* **130**, 124502 (2009).

[49] Spreafico, C. & VandeVondele, J. The nature of excess electrons in anatase and rutile from hybrid dft and rpa. *Phys. Chem. Chem. Phys.* **16**, 26144–26152 (2014).

[50] Deskins, N. A. & Dupuis, M. Electron transport via polaron hopping in bulk $\mathrm{Tio_2}$: A density functional theory characterization. *Phys. Rev. B* **75**, 195212 (2007).

[51] Wang, T. *et al.* Anisotropic d–d transition in rutile tio2. *The Journal of Physical Chemistry Letters* **0**, 10515–10520 (0).

[52] Sahasrabudhe, G., Krizan, J., Bergman, S. L., Cava, R. J. & Schwartz, J. Millionfold increase of the conductivity in tio2 rutile through 3% niobium incorporation. *Chemistry of Materials* **28**, 3630–3633 (2016).

[53] Deskins, N. A., Rousseau, R. & Dupuis, M. Distribution of ti3+ surface sites in reduced tio2. *The Journal of Physical Chemistry C* **115**, 7562–7572 (2011).

[54] Hao, X., Wang, Z., Schmid, M., Diebold, U. & Franchini, C. Coexistence of trapped and free excess electrons in $\mathrm{srtio_3}$. *Phys. Rev. B* **91**, 085204 (2015).

[55] Reticcioli, M., Setvin, M., Schmid, M., Diebold, U. & Franchini, C. Formation and dynamics of small polarons on the rutile $\mathrm{tio_2}(110)$ surface. *Phys. Rev. B* **98**, 045306 (2018).

[56] Guandalini, A., D'amico, P., Ferretti, A. & Varsano, D. Efficient gw calculations in two dimensional materials through a stochastic integration of the screened potential. *npj Comput. Mater.* **9**, 44 (2023).