GOALI: Elements: Atomistic Tool for Open, Machine-learned Identification and Characterization (ATOMIC)

Overview

Atomistic simulations provide tremendous insights into the atomic origins of phenomena in materials science, chemistry, biology, and myriad other fields. In order to reveal these insights, atomistic datasets must be analyzed to identify formation and evolution of crystal, molecular, or amorphous structures and their defects of relevance to phenomena such as phase transformations, thermal transport, and protein folding. Unfortunately, the materials science community lacks a methodology for atomic structure analysis which is readily extensible to new kinds of structures and is robust to thermal noise in atomic positions. The goal of this project is to develop such a method through innovative software that enables a machine-learning (ML)-based atomic structure analysis paradigm. Under this project, four software packages forming the Atomistic Tool for Open, Machine-learned Identification and Characterization (ATOMIC) will be developed for the research community: the ATOMIC analyzer which researchers use to perform structure analysis on atomistic datasets; the ATOMIC machine learning platform which constructs ML classifiers used for structure analysis in the analyzer; the ATOMIC workflow manager which manages computational workflows during classifier training; and the ATOMIC web portal where users access and contribute to a growing database of structure classifiers. With our GOALI partner OVITO GmbH, a graphical user interface for ATOMIC will be implemented in the widely used OVITO visualization tool, which has 45,000 active users in the research community analyzing atomistic datasets. The overall ATOMIC framework enables a revolutionary data-driven and crowd-sourced approach to structure analysis that is uniquely flexible and extensible in comparison to existing analysis tools.

Intellectual Merit

ATOMIC will trailblaze a new paradigm for development and extension of cyberinfrastructure (CI) which seeks to meet the growing and evolving needs of the scientific community. In this paradigm, data-driven approaches are adopted that enable automated subroutines which extend the capability of a research tool given a new user request, while utilizing diverse high-performance computing resources provided by the community. The ATOMIC tool itself will revolutionize structure analysis of atomistic datasets by providing the first broadly extensible analysis framework. This will have tremendous impact on researchers in the areas of materials, chemical, biological, and geological sciences which greatly lack in analysis capability. This impact is achieved by using ML to construct classifiers for each structure class, rather than ad hoc and heuristic rules. The ATOMIC ML software package will be a prototype for execution of data-driven classification with scientific data by automating feature selection and classifier design in a flexible, extensible, and efficient framework. Finally, the ATOMIC workflow manager software package will be a generic tool for managing large-scale computational workflows across diverse and varied high-performance computing resources, a valuable capability for many research areas.

Broader Impacts

The new capabilities provided by ATOMIC will revolutionize the impact of atomistic simulations on research and applications in energy, defense, transportation, and beyond. Through the organization of symposia and workshops, this project will promote and foster a new community of researchers focused on advancing atomic structure analysis with the goal of extracting new fundamental insights from atomistic simulations. This project provides a unique opportunity to bridge the materials science, machine learning, and software engineering communities, serving as an exemplar for integrating domain knowledge and data science in CI development. A STEM outreach program employing high school interns and undergraduate researchers, especially from underrepresented groups, will help foster the next generation of computational scientists.

1 Project Motivation and Impact

1.1 Science-driven

Atomistic modeling tools offer tremendous insight into the mechanistic origins of the behaviors of substances. Such techniques, including density functional theory (DFT) and molecular dynamics (MD), evolve atomic structures in time according to interatomic forces and externally imposed conditions (stress, temperature) [1–3]. Phenomena such as phase transitions [4], deformation and damage [5, 6], thermal [7], mass [8], and electronic transport [9, 10], adhesion and wear [11], and folding and restructuring of molecules [12]—to name a few—can be studied with full atomic detail and with few assumptions/approximations about underlying physics. The use of atomistic modeling has become widespread in science and engineering due to its impact on energy, defense, transportation, and aerospace applications [13]. Fig. 1 shows that the number of publications per year on atomistic modeling has been growing exponentially since 1990, reaching $\sim$15,000 by 2020. By another measure, the open-source MD code LAMMPS was downloaded 405,000 times between 2004 and 2021 [14]. Despite widespread success, one major obstacle persists in the analysis of atomistic simulation datasets: *interpretability*. Atomic systems typically arrange themselves into different types of structures comprised of numerous phases, defects, and molecules (a few examples are shown in Fig. 2). At nonzero temperatures atoms "wiggle and jiggle" about, introducing thermal noise into the structures. As a result, identifying which structure(s) each atom is associated with is not a trivial task. The basic scientific question underlying atomic structure analysis is: *given a set of thermally disturbed atomic positions, how can we accurately and efficiently identify the structure(s) associated with each atom?*

To further demonstrate the challenge and impact of atomic structure analysis, we have assembled in Fig. 2 a list of 35 common defects in metals for which there are currently no analysis tools available. These defects are relevant to many phenomena in science and engineering. In our research here, we will focus on void nucleation by vacancy condensation as an exemplar. Vacancy condensation is a critical aspect of fracture [15], creep loading [16], hydrogen embrittlement [17], and irradiation damage [18], but is extremely difficult to study because of the inability to identify and track vacancies, impurity interstitial atoms, and self-interstitial atoms (SIAs) in MD simulations. As a result, the rates of vacancy generation during plastic deformation [19] and subsequent condensation into voids are poorly understood. Using the software solutions we outline below, we will quantify these rates for the first time via direct MD studies. As another example of the challenge at hand, consider the 1,100 unique crystal structures which the AFLOW Encyclopedia of Crystallographic Prototypes [20] has identified. Existing crystal structure identification tools are able to identify 8 elemental structures [21], 5 ice compounds [22], and 7 ordered compounds [23], leaving 1,080 crystal structures that cannot be identified. This does not even consider the many possible defects in these structures or the ever-expanding library of two-dimensional materials and molecular structures. Clearly the current approaches to structure identification are inadequate for the broad needs of the atomistic science community. Key insights that could lead to revolutionary discoveries in materials science, chemistry, and medicine remain locked away inside of currently uninterpretable atomistic data. Filling this capability gap can revolutionize atomistic modeling and its research impact across the globe.

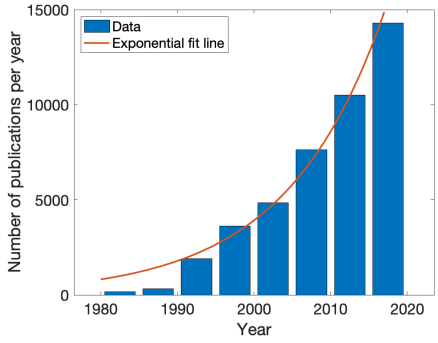


Figure 1: Number of scientific journal publications per year mentioning "atomistic simulations" or "molecular dynamics" (data from Web of Science) with exponential fit.

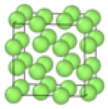
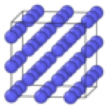
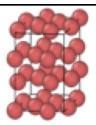
		Vacancy	Self-interstitial atom	Impurity interstitial atom	Stacking fault	Twin bounda
	FCC	1	1 [100]-dumbbell	2 tetrahedral, octahedral	1 extrinsic	----
	BCC	1	2 [110]-dumbbell, [111]-dumbbell	2 tetrahedral, octahedral	0	1
	HCP	1	8 tetrahedral, octahedral, basal tetrahedral, basal octahedral, basal crowdion, non-basal crowdion, basal split dumbbell, *c* split dumbbell	6 tetrahedral, octahedral, basal tetrahedral, basal octahedral, basal crowdion, non-basal crowdion	2 extrinsic, intrinsic	7 (2̄111), (2̄112), (2̄113), (2̄1 (1̄011), (1̄012), (1̄013)

Figure 2: 35 common defects in body-centered cubic (BCC), face-centered cubic (FCC), and hexagonal close-packed (HCP) crystals which DO NOT currently have analysis tools available. Values in the table denote the number of unique variants for each defect type and crystal structure. Top snapshots show example defects in FCC metals.

1.2 Innovation

Two major challenges stand in the way of a comprehensive analysis tool for atomic structures. The first challenge is identifying discriminating features which differentiate structures from each other. This is currently done by hand (e.g., phase A has 4 nearest neighbors, but phase B has 6), which is a laborious, slow, and often ineffective process. The second challenge is identifying what specific structures are of interest to the community. Given the ever-expanding, vast array of substances studied by science communities, it is not possible for one (or even a small group of) researchers to enumerate a comprehensive list of structures. The *hypothesis* driving our proposed work is that these challenges can be overcome through software which enables a *data-driven, crowd-sourced approach to atomic structure analysis*. Using a data-driven approach automates identification of discriminating features through the use of machine learning (ML). And crowd-sourcing makes it possible to appeal to the entire community for input on what structures are important for the advancement of science and engineering knowledge.

Combining these concepts, we propose a *software-enabled paradigm for atomic structure analysis* that takes user requests for structures and uses ML to construct classifiers for identifying these structures. We call this approach ATOMIC: Atomistic Tool for Open, Machine-learned Identification and Characterization. To our knowledge, this would be a pioneering data-driven, crowd-sourced tool for analysis of science datasets. Our success with ATOMIC can therefore trailblaze a paradigm shift in the analysis of experimental and simulation datasets across disciplines.

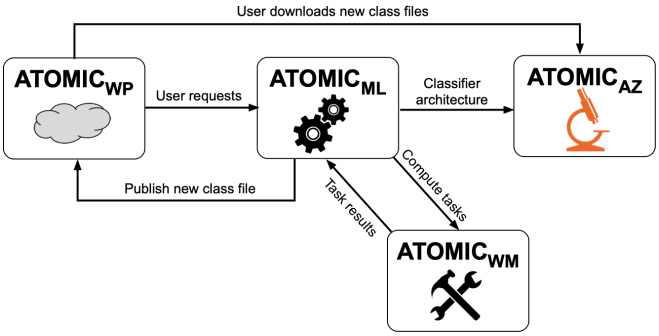


Figure 3: Workflow among the four ATOMIC software packages.

The most significant challenge to realizing ATOMIC is the need for efficient, scalable, and automated computation driven by novel algorithms and software. Specifically, software is required which (1) accepts user requests for new structures, (2) automatically generates atomistic datasets for the requested structures, (3) trains flexible and scalable ML classifiers for identifying these structures, and (4) posts the resulting classifier to a searchable online database so other users can rapidly discover it. To achieve these objectives, innovations are required in ML and workflow management. While a few recent efforts have shown the feasibility of ML-based structure analysis with a fixed library of structures [24, 25], constructing an ML framework which adapts continuously to an ever-evolving library of structures is a major challenge; there are no "plug and play" solutions to this challenge. Secondly, our vision of ATOMIC which ensures sustainability is the usage of computing resources made available by sponsors, users, and enthusiasts. Middleware is necessary to manage workflows on such a disparate, distributed pool of compute resources. In fact, the need for workflow management middleware goes well beyond ATOMIC; other examples include OpenKIM [26], Folding@Home [27], and QCArchive [28]. Another major innovation of ATOMIC is the use of a scalable, domain-agnostic middleware workflow management platform which handles disparate computational tasks and computing resources.

2 Cyberinfrastructure Plans

2.1 Project plans, and system and process architecture

Our *goal* is to develop and implement ML-based *classifiers* which are able to distinguish among a set of atomic *structure classes*. Towards this goal, ATOMIC will consist of four interrelated software packages (see Fig. 3), each serving a distinct function:

- ATOMIC Analyzer, $\text{ATOMIC}_{\text{AZ}}$: The *workhorse* of ATOMIC, a user-facing analysis tool for atomistic datasets
- ATOMIC Machine Learning Framework, $\text{ATOMIC}_{\text{ML}}$: The *brains* of ATOMIC, used to generate classifiers for each atomic structure class
- ATOMIC Workflow Manager, $\text{ATOMIC}_{\text{WM}}$: The *foreman* of ATOMIC, used to manage and allocate computing resources for $\text{ATOMIC}_{\text{ML}}$
- ATOMIC Web Portal, $\text{ATOMIC}_{\text{WP}}$: The *commons* of ATOMIC, where users access a database of available structure classifiers

While our approach will heavily leverage existing software, it is important to emphasize that the ATOMIC framework is currently impossible to attain because of the lack of appropriate software and algorithms. Our team is uniquely qualified for such an effort (see Section 2.3).

Users will interact most commonly with $\text{ATOMIC}_{\text{AZ}}$, the user-facing package wihch analyzes datasets and identifies structure class(es) for each atom on local compute resources. To promote widespread adoption, we will implement $\text{ATOMIC}_{\text{AZ}}$ for use within the open-source OVITO Basic software [29] along with developers at OVITO GmbH, our GOALI partner. OVITO Basic is a tremendously popular, freely-available tool, with 40,000 active users and $\sim$150 journal articles published each month using OVITO (data from OVITO GmbH). If a structure of interest is not available in $\text{ATOMIC}_{\text{AZ}}$, users will search the structure database in $\text{ATOMIC}_{\text{WP}}$ for an existing structure class file which can be loaded into $\text{ATOMIC}_{\text{AZ}}$. If no such file is available, users can request new structures be added via $\text{ATOMIC}_{\text{WP}}$. The overall structure classification framework is designed by $\text{ATOMIC}_{\text{ML}}$, which takes requested structures, generates training/testing data, and then trains a suitable ML classifier that integrates with the existing library. This process requires large-scale computation during data generation and classifier training. $\text{ATOMIC}_{\text{WM}}$'s job is to efficiently manage these computations across diverse compute resources made available by sponsors, users, and enthusiasts. Once a new classifier is successfully trained by $\text{ATOMIC}_{\text{ML}}$, the resulting class file will be posted to $\text{ATOMIC}_{\text{WP}}$ so that any user can access it and load it into $\text{ATOMIC}_{\text{AZ}}$.

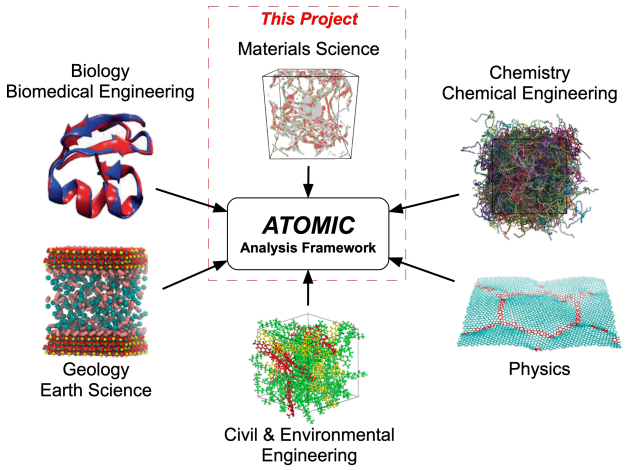


Figure 4: Our long-term vision is for ATOMIC to be an analysis tool for the entire research community. This project initializes that vision with a more narrow focus on materials science. Graphics from References [12, 30–34].

The long-term vision is for ATOMIC to have a *fully automated* workflow, whereby user requests to ATOMIC$_{\text{WP}}$ initiate a set of computations coordinated by ATOMIC$_{\text{ML}}$ and executed by ATOMIC$_{\text{WM}}$, culminating in the publication of a new class file on ATOMIC$_{\text{WP}}$. A fully automated workflow is necessary to make ATOMIC massively scalable with maximal impact on the community. Furthermore, we envision ATOMIC as a revolutionary tool for any science community utilizing atomistic simulations (see Fig. 4). Clearly such a revolutionary, expansive vision cannot be realized in a single, 3-year grant. Our plan for achieving this vision is to execute two developmental stages. Stage 1—to be executed under this project—will develop a semi-automated workflow prototyped on Rutgers/NSF clusters and focused on the needs of the metals research community by providing classifiers for all defects in Fig. 2. We will then apply these classifiers to our exemplar problem of void nucleation by vacancy condensation under scenarios relevant to spall fracture and irradiation damage. This makes our proposed work here primarily align with the *Division of Materials Research* under the Directorate for Mathematical and Physical Sciences and the *Division of Civil, Mechanical and Manufacturing Innovation* under the Directorate for Engineering. The resulting atomistic analysis tool will have unprecedented capability and versatility, but extension to new structures will require some manual effort. After Stage 1 demonstrates promising results, in Stage 2 we will pursue additional funding support—possibly through a CSSI Frameworks proposal—for the fully automated and scaled-up version of ATOMIC which will be designed to reach a much broader research community.

2.1.1 ATOMIC$_{\text{AZ}}$: The ATOMIC Analyzer

Design: ATOMIC$_{\text{AZ}}$ will be developed in the same style as similar, non-data-driven analysis tools, such as adaptive common neighbor analysis (aCNA) [35], polyhedral template matching (PTM) [21], and VoroTop [36]. These are standalone, command-line C++ codes which take atomic positions and species as inputs and return structure assignments to each atom. Similar to PTM, we will develop ATOMIC$_{\text{AZ}}$ to be agnostic of the specific compute environment and platform so that it is easily deployed in OVITO, LAMMPS [37, 38], and other modeling/visualization packages.

The architecture of ATOMIC$_{\text{AZ}}$ derives from the ML architecture developed by ATOMIC$_{\text{ML}}$ in terms of the underlying ML techniques and formulations (more details about ATOMIC$_{\text{ML}}$ in the next section). To apply the tool to an ever-expanding set of classes, ATOMIC$_{\text{AZ}}$ will accept new structure class files as inputs. Structure class files contain all necessary information to fully define an ML classifier for a specific atomic structure, and can be obtained by the user via the web portal, ATOMIC$_{\text{WP}}$. For example, a class file may contain information on the type of ML classifier used for a given structure class (e.g., neural network, SVM), the ML classifier architecture, the set of atomic descriptors which are used to characterize the local atomic environments, and values for ML hyperparameters resulting from training in ATOMIC$_{\text{ML}}$.

A critical aspect of ATOMIC$_{\text{AZ}}$ is scalability and computational efficiency, such that large-scale datasets (e.g., $> 10^6$ atoms) can be efficiently analyzed. This will be achieved by utilizing state-

of-the-art open-source analysis libraries (e.g., TensorFlow [39], Voro++ [40]), designing efficient algorithms for computations, and by implementing in a low-level language (e.g., C++).

Most users are expected to interact with $\text{ATOMIC}_{\text{AZ}}$ via the graphical user interface (GUI) in OVITO Basic. Hence, developing an intuitive, effective GUI is critical to the success and adoption of ATOMIC. For this reason, we will directly partner with OVITO GmbH, developer of OVITO, so that ATOMIC developers and user feedback have direct input on the GUI design. The overall design will be similar to existing GUIs for structure analysis tools within OVITO. However, ATOMIC will offer a broader set of capabilities than any other atomic structure analysis tool. The primary goal of this project will to enable identification of the metal defects listed in Fig. 2, while establishing a framework that can reach well beyond that list. The OVITO GUI will be able to directly access and communicate with the ATOMIC web portal, $\text{ATOMIC}_{\text{WP}}$, via it's API (see Section 2.1.4). This enables users to search and download from the ATOMIC database of structure classes within OVITO, providing a rapid, easy user experience. Users will be able to submit requests for new classes within OVITO. We emphasize that OVITO GmbH offers an open-source version of OVITO which will house the ATOMIC GUI, making ATOMIC freely available to all users.

Development: $\text{ATOMIC}_{\text{AZ}}$ is comprised of two core modules, as shown in Fig. 5(b): DescriptorCalculator and ClassifierEvaluator. The DescriptorCalculator module will take raw atomic data and compute a set of descriptors for the atomic environment surrounding each atom. Details on these descriptors and their computation are provided in Section 2.1.2. The ClassifierEvaluator module takes the set of atomic descriptors for each atom as input and evaluates the appropriate ML classifier, returning a set of atomic structure classes for each atom as output.

Testing: Testing will be based on training and testing datasets used by $\text{ATOMIC}_{\text{ML}}$. The final results for classification accuracy obtained by $\text{ATOMIC}_{\text{ML}}$ will be documented. $\text{ATOMIC}_{\text{AZ}}$ should exactly reproduce these results. A regression test library will be developed on the basis of these datasets and results, which will be applied to $\text{ATOMIC}_{\text{AZ}}$ source code during development.

To further "stress test" $\text{ATOMIC}_{\text{AZ}}$, a diverse set of test cases will be constructed to demonstrate robust classifier performance and $\text{ATOMIC}_{\text{AZ}}$ execution across research problems. Examples of such test cases include: nanocrystalline systems, multi-phase systems, porous systems, and systems strained/deformed in various ways. These stress tests will also demonstrate robustness in the OVITO GUI implementation.

2.1.2 $\text{ATOMIC}_{\text{ML}}$: The ATOMIC Machine Learning Framework

Design: $\text{ATOMIC}_{\text{ML}}$ is the brains of ATOMIC, utilizing established ML tools (e.g., TensorFlow) to establish a framework for identifying atomic structures. It has three design requirements: (R1) functionality–its ability to perform high-quality (e.g., based on accuracy, AUC, recall) classification of atomic structures; (R2) computational efficiency–in terms of both classifier training and evaluation;and (R3) extensibility–its capability to accommodate, with minimal changes, future expansions to new atomic structure classes.

Achieving R1-R3 entails a number of unique data science challenges that go beyond existing software capabilities. For instance, classifiers that are tailored towards a particular classification task can require major changes to their architectures and/or inputs when additional classes are introduced, making them less effective for ever-expanding classification problems (such as here). In response, our design concept, supported by preliminary experimental evidence shown below, is that cascade classifiers (CCs)—a class of ensemble learning methods which have recently gained a wide attraction in ML research [41–46]—combined with a well-designed feature selection mechanism, effectively meet R1-R3. While the basic ML techniques we will employ are well established, an efficient scalable framework which enables an adaptive classification toolkit will be a key innovation of the project.

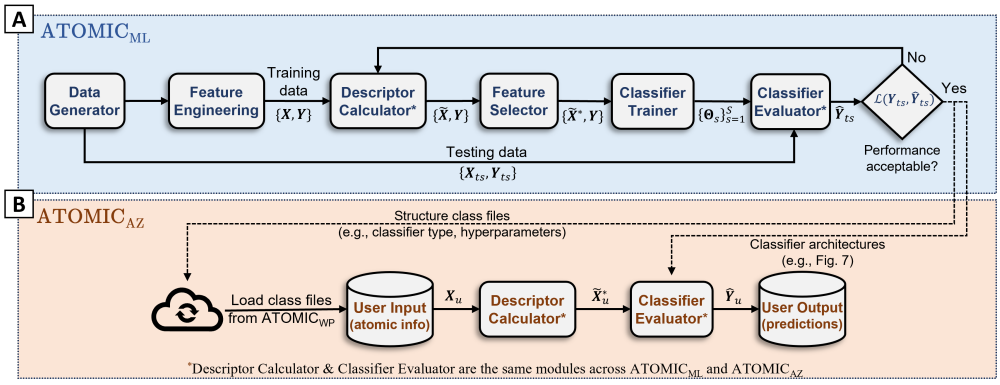


Figure 5: High-level flow of ATOMIC$_{ML}$ (Panel A) and its coupling with ATOMIC$_{AZ}$ (Panel B).

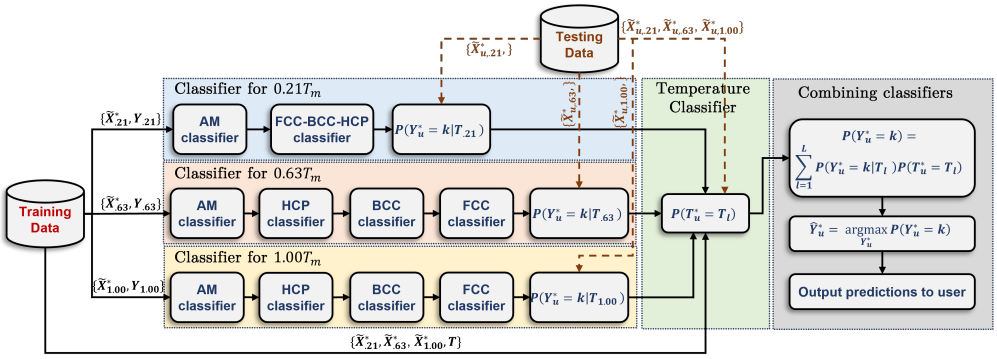


Figure 6: Preliminary results: An ensemble of cascade classifiers to distinguish FCC, BCC, HCP, and AM crystals of Fe, at *any* arbitrary temperature.

Fig. 5 shows the high-level software design of ATOMIC$_{ML}$, and how it is coupled with ATOMIC$_{AZ}$. ATOMIC$_{ML}$ starts by running MD simulations which will be used to generate an initial dataset of n atoms. The dataset comprises the labels of the atoms (e.g., atom phase, defect type), stored in the n-dimensional vector $\mathbf{Y}$, along with a set of p features which characterize the local atomic environment surrounding each atom. Many descriptors are possible [47] including: atomic density distributions [48, 49], atomic density correlations [50], atomic cluster expansions [51, 52], and topological features such as graph [35, 53, 54], Voronoi [36, 55], or other polyhedral constructions [21, 56]. In our preliminary work, we have focused on topological attributes of the Voronoi and nearest neighbor polyhedra surrounding each atom, such as their volume, area, vertex distances, etc. Our choice of topological features flows from recent work [21, 36, 55], as well as our preliminary results (shown below), confirming that topology-guided features can effectively *and* compactly characterize atomic environments. We denote this initial $n \times p$ matrix of features as $\mathbf{X}$. An automated *feature engineering* step will then augment $\mathbf{X}$ by constructing exogenous information which may possess additional explanatory power. Examples include deviations from perfect crystal structures, measures of centrality, dispersion, modality, higher order moments, etc. This forms the augmented $n \times m$ matrix $\tilde{\mathbf{X}}$, where $\mathbf{X} \subset \tilde{\mathbf{X}}$, and $m > p$. Finally, an automated *feature selection* step will efficiently find a minimally-sized, information-rich subset of $\tilde{\mathbf{X}}$, call it $\tilde{\mathbf{X}}^*$, which is of dimension $n \times q$, thereby reducing model complexity, ensuring input orthogonality, and curtailing overfitting risks. To minimize re-training, we will initially follow a *filter* selection mechanism [57], wherein features are filtered based on computationally cheap measures of information gain (e.g., histogram distances, relief and Fisher scores).

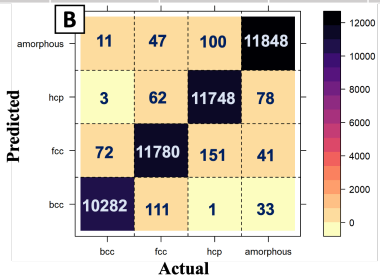
Given $\tilde{\mathbf{X}}^*$, a cascade classifier will be trained to perform the classification task. The main distinguishing aspect of CC relative to mainstream ensemble learning (e.g., AdaBoost, model averaging) is that the individual classifiers within the cascade can operate in a variety of modes: in-parallel (individual classifiers are independent, whereas predictions are pooled based on some weighing scheme), in-series (individual classifiers are connected via a set of accept/reject thresholds), or a combination thereof [41]. This flexibility in design is key to achieve R1-R3. For instance, parallelization of the training load for individual classifiers enables leveraging parallel computational resources. This flexibility also reduces the need to fully re-train the classifier when a new class is introduced, i.e., if a CC is initially designed to separate $k-1$ classes, then it suffices, in many cases, to simply append an additional member to the cascade in order to accommodate a kth class, while maintaining a satisfactory level of prediction accuracy.

A

$\text{ATOMIC}_{\text{ML}}$ applied to four classes (fcc, bcc, hcp, amorphous)

Method	Accuracy	Precision	Recall	F-score
$\text{ATOMIC}_{\text{ML}}$	**98.469%**	**98.477%**	**98.492%**	**98.485%**
aCNA	89.926%	94.538%	89.784%	90.484%

B



C

Extending $\text{ATOMIC}_{\text{ML}}$ to a new class (Vacancies, total # of classes = 5)

Method	Accuracy	Precision	Recall	F-score
$\text{ATOMIC}_{\text{ML}}$	**98.390%**	**98.573%**	**98.451%**	**98.512%**
aCNA	Doesn't exist	Doesn't exist	Doesn't exist	Doesn't exist

Figure 7: (a) Table shows the performance of $\text{ATOMIC}_{\text{ML}}$ vs. aCNA, in terms of various classifier quality metrics. (b) Confusion matrix showing match between actual and predicted structure type. (c) Performance of $\text{ATOMIC}_{\text{ML}}$ when extended, with minor adjustments, to the new vacancy class (now total # of classes = 5), with fairly imbalanced data. Currently, no existing tool can identify vacancies.

The training step will result in a set of hyperparameters $\boldsymbol{\Theta}$, which will be passed, along with the reserved test set $\{\mathbf{X}_{ts}, \mathbf{Y}_{ts}\}$, to the ClassifierEvaluator module where atomic structure predictions, denoted by $\hat{\mathbf{Y}}_{ts}$, will be computed and then compared, using various loss metrics, $\mathcal{L}(\cdot,\cdot)$, to the actual labels $\mathbf{Y}_{ts}$. If the test is not passed, the procedure is re-iterated to re-construct $\tilde{\mathbf{X}}^*$ by adding/dropping additional features in order of feature importance. On the other hand, if the test is passed, the resulting CC architecture is implemented for use in $\text{ATOMIC}_{\text{AZ}}$, along with publication of associated structure class files on $\text{ATOMIC}_{\text{WP}}$.

For validation, we conducted a case study where the goal is to distinguish FCC, BCC, HCP, and amorphous (AM) crystals of Fe at *any* temperature. We first performed MD simulations with $n \approx 10^4$ atoms (evenly distributed across the 4 classes) to generate training and testing data at three temperatures: $0.21T_m$, $0.63T_m$, and T_m, where T_m is the melting temperature. We then designed an ensemble of CCs (architecture shown in Fig. 6) which comprises one CC for each temperature, combined with a temperature classifier. The design was motivated by our observation that $\tilde{\mathbf{X}}^*$ is temperature-dependent, i.e., different sets of features are relevant at different temperatures. The temperature classifier, on the other hand, predicts the temperature level at which the user-input data is generated. A simple probabilistic pooling step produces the final set of predictions.

Feature selection was performed via the average histogram distance [58]. As an example, $\tilde{\mathbf{X}}^*$ for the $0.21T_m$ classifier had $q = 26$ features, including few Voronoi attributes like polyhedron volume and area, as well their 30-atom spatial averages, and other "engineered" summary statistics of such features. The classifier used was a Gaussian mixture model, primarily selected for its ability to produce probabilistic predictions and low training requirements. The results are shown in Fig. 7A-B, suggesting a superior performance, in terms of average predictive performance metrics, relative to aCNA [35], which is OVITO's default atomic structure classifier.

For further validation, we designed additional simulations to introduce a new structure class to our classifier: vacancies, a defect where an atom is missing from a lattice site (see Fig. 2). *No existing tools can identify vacancies.* Since vacancies are defects, they are typically a "minority class."

$\text{ATOMIC}_{\text{ML}}$ only required adding another classifier member, trained on the vacancy data, to be added to the CC, without the need to re-train the other CC members. The Table in Fig. 7C shows that the performance remains very competitive.

This case study demonstrates how CCs coupled with a feature selection, can be well-suited for our design requirements R1-R3, wherein (i) adding or dropping a class, or a new temperature level, can be easily accommodated without redesigning or fully re-training the classifier; (ii) the choice of the individual classifiers is flexible to balance efficiency and accuracy as needed; and (iii) large portions of the training load of the CC in Fig. 6 can be parallelized.

Development: Since $\text{ATOMIC}_{\text{ML}}$ needs to compute descriptors and evaluate classifiers in the same way that $\text{ATOMIC}_{\text{AZ}}$ does, the DescriptorCalculator and ClassifierEvaluator code modules can be shared by the two packages. Again, we can decompose the underlying software into a set of coupled modules, following the workflow in Fig. 5. The first module is DataGenerator, which generates MD simulation datasets, used for classifier training and testing. The second module is FeatureEngineering which constructs additional exogenous variables using the raw MD datasets. The third module is FeatureSelector which automatically determines a sparse set of features which possesses high explanatory power. The fourth and fifth modules are TrainClassifier and TestClassifier where the classifier design is determined, and the training and testing are performed. For ease of development, initial implementation of $\text{ATOMIC}_{\text{ML}}$ will be in R and/or the Python implementation of TensorFlow. These interpreted codes will not provide necessary performance for production runs, so the code will eventually be ported over to the C++ TensorFlow library.

Testing: During the development of $\text{ATOMIC}_{\text{ML}}$, several sets of standardized test cases will be developed (e.g., see preliminary results) which will baseline the performance and computation time of $\text{ATOMIC}_{\text{ML}}$. These test cases, comprised of training and testing datasets and classifier performance metrics, will be used to construct a regression test library for $\text{ATOMIC}_{\text{ML}}$.

2.1.3 $\text{ATOMIC}_{\text{WM}}$: The ATOMIC Workflow Manager

Design: $\text{ATOMIC}_{\text{WM}}$ will be built using RADICAL-Cybertools (RCT), software systems designed and implemented using a building blocks approach. Each system is designed with well-defined entities, functionalities, states, events and errors. Ref. [59] discusses three existing RCT systems: RADICAL-Ensemble Toolkit (EnTK) [60]), RADICAL-Pilot (RP) [61] and RADICAL-SAGA (RS) [62] used to support high-performance and distributed computing (HPDC).

Individual RCT components are designed to be consistent with a four-layered view of distributed systems for the execution of scientific workloads and workflows on HPDC resources. Each layer has a well-defined functionality and an associated "entity". The entities are **workflows** (L4)(or applications) at the top layer and resource specific **jobs** (L1) at the bottom layer, with **workloads** (L3) and **tasks** (L2) as intervening transitional entities in the middle layers.

RADICAL-Pilot (RP) is a scalable Python implementation of the pilot paradigm and architectural pattern [63]. Pilot systems enable users to submit pilot jobs to computing infrastructures and then use the resources acquired by the pilot to execute one or more tasks. These tasks are directly scheduled via the pilot, without having to queue in the infrastructure batch system. The defining capability is to decouple resource acquisition from task execution. Pilot systems allow for queuing a single job via the batch system and, once this job becomes active, it executes a system application that enables the direct scheduling of tasks on the acquired resources, without waiting in the batch system's queue. In this way, pilot systems can enable high throughput computing (HTC) on infrastructures designed to enable high performance computing (HPC).

RP is optimized for HPC resources, enabling heterogeneous workloads of one or more scalar, MPI, OpenMP, multi-process, and multi-threaded tasks. These tasks can be executed on CPUs, GPUs and other accelerators, on the same pilot or across multiple pilots. Each task is a program,

running as a self-contained executable and not as a function, method, thread or process of a parallel application.

The design of RP and other RCT conform to the principles of self-sufficiency, interoperability, composability and extensibility. RP is: self-sufficient because it independently implements the necessary and sufficient set of functionalities for its entities; interoperable in terms of type of task, resource, and execution paradigm; and extensible as new properties can be added to the pilot, unit and resource descriptions, and more functionalities can be implemented for these entities. Currently, composability is partially designed and implemented: while the API can be used by both users and other systems to describe generic tasks for execution, RP requires RADICAL-SAGA to interface to HPC resources.

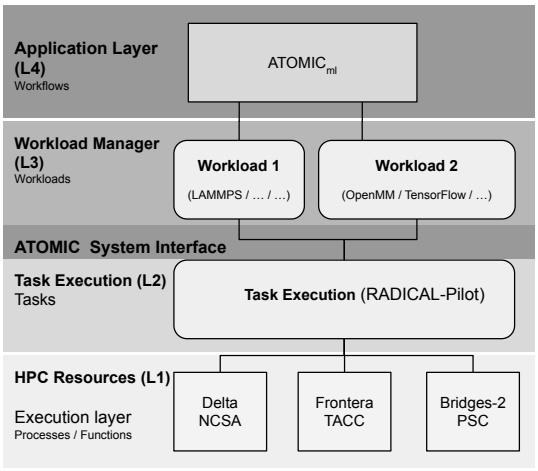


Figure 8: Design of the ATOMIC Workflow Manager Component. ATOMIC$_{WM}$ uses RADICAL-Cybertools – a proven production and scalable set of middleware building blocks that are used to manage and execute workflows across HPC and Cloud systems.

Each cybertool is an independent system that can also be integrated with other systems (RCT or otherwise) to form tailored middleware solutions. For example, several independent communities directly utilize RADICAL-SAGA alone, some RP. Other communities integrates all RCT with or without third-party systems to support the execution of diverse types of scientific workflows. Thus, RCT are not posed to replace existing workflow systems, nor they are, as a whole, an end-to-end workflow system: RCT's novelty is to enable the integration across systems independently developed and not necessarily designed to integrate. Crucially, this could be at different levels: workflow, workload and computing frameworks.

An ecosystem in which end-to-end workflow systems and building blocks coexist and, when useful, are integrated helps to avoid both vendor lock-in and fragmentation. Such an ecosystem allows scientists with specific and stable requirements to use an end-to-end system while others to aggregate existing capabilities into tailored solutions.

As building blocks, RCT offer several benefits [59]: the most relevant is isolating developers of workflow tool or scientists (L4) from job management (L1), task management (L2), and workload management (L3), letting L4 users exclusively focus on workflow description and application logic. While this isolation is offered by other systems, RCT is agnostic towards which software and systems are integrated at each layer L1–4.

Development: As shown in Fig. 8 we will bring the building block philosophy and advantages to ATOMIC. The most important decoupling is between the application and workload layers on the one hand, and task execution and resource layers on the other. In practice, this results in the ability for ATOMIC$_{WM}$ to execute tasks on different (heterogeneous) resources independent of the details of the application and workloads. RADICAL-Cybertools however have to be ported to the set of resources that ATOMIC will use (L1). The first development track will address this requirement. Specifically, this requires ensuring RADICAL-SAGA and RADICAL-Pilot work are engineered to work on a range of HPC platforms – campus clusters to NSF leadership platforms.

ATOMIC$_{WM}$ will serve as the primary coordination and management component, which will receive a mixed workload of HPC simulations and ML tasks and pass to the execution layers (L2 and L1). Thus, a second development track will be to identify and implement appropriate abstrac-

tions (at level L3) to be able to express the coupling, dependencies and communication between distinct tasks comprising the mixed workload. We will use the existing EnTK API [59] as a starting point. EnTK will be used to implement the capability to manage the mixed HPC simulations and ML workloads [64]. EnTK sends the sets of tasks that are ready to run to the RADICAL-Pilot (L2) which will manage the efficient resource utilization and task execution. Also, ATOMIC$_{WM}$ will leverage the RADICAL internal database to acquire and store task execution and state information. Put together, these two tracks will allow ATOMIC$_{WM}$ to support many applications and workloads utilize a diverse range of HPC resources while being agnostic of specific details of L4 and L3 (i.e., how the applications are expressed).

Testing: All RADICAL tools development employ unit, module and integration testing protocols. In addition, we will extend the Continuous Integration infrastructure to test deployments on production platforms (such as XSEDE, and University resources).

2.1.4 ATOMIC$_{WP}$: The ATOMIC Web Portal

Design: The web portal serves four key functions for ATOMIC: (1) act as the main interface for the community to learn about ATOMIC, (2) provide a searchable database for accessing structure class files for atomic structures and associated training/testing data, (3) accept user requests for new structures to be added to ATOMIC, and (4) provide a forum for discussion among users and developers. Furthermore, to enable broader access to the database of available structure classes, an API will be developed so that ATOMIC$_{WP}$ can be readily accessed by other software applications, such as OVITO and LAMMPS. This API will be leveraged by the OVITO-ATOMIC interface.

The web page will contain pages with the following content: *Get ATOMIC* – information on ways to access ATOMIC source codes (e.g., via GitHub and OVITO); *Documentation/Tutorials* – searchable wiki-style documentation, guided examples, and video tutorials connected with ATOMIC's YouTube page; *Structure Database* – searchable, sortable, and filterable database of structure classes; *User Requests* – templated form where users can request new structure classes to be added to ATOMIC; *Forum/Discussion Board* – community resource where researchers can post/answer questions, share findings and insight, and discuss issues related to analyses of atomic datasets. In addition, within the Structure Database page, there will be a browser-integrated OVITO viewer so that users can visualize and manipulate each structure, leveraging ongoing capability enhancements at OVITO GmbH.

Development: The main interface for ATOMIC$_{WP}$ will be coded in HTML and PHP, and have a MySQL server on the back end, which will be also used to store the searchable database of structure classes. The custom forms with user input will be processed and stored into MySQL database tables. Drupal CMS minisite will be deployed for user discussion forums. Users will be able to create accounts online, login, or put postings anonymously.

ATOMIC$_{WP}$ will support a public interface to enable querying of the structure database. We will consult with the NSF Science Gateways Institute for the design of science gateways (see letter), and employ best practises in API design and usability assessment. We will use OpenKIM's [26] interface style and structure as a starting evaluation point (see letter of collaboration), but the production interface specification will be derived from our requirements elicitation and available data access and storage methods.

Testing: The web portal will be stress tested by "clicking through" all links and features, including posting to forum pages, submission of user requests, viewing documentation/tutorials, and searching/manipulation of the structure class database.

2.1.5 Integration and Validation

Integration of ATOMIC involves activating the overall software workflow among the packages as depicted in Fig. 3. There are two Integration Tasks: (1) integration of ATOMIC$_{WP}$ with

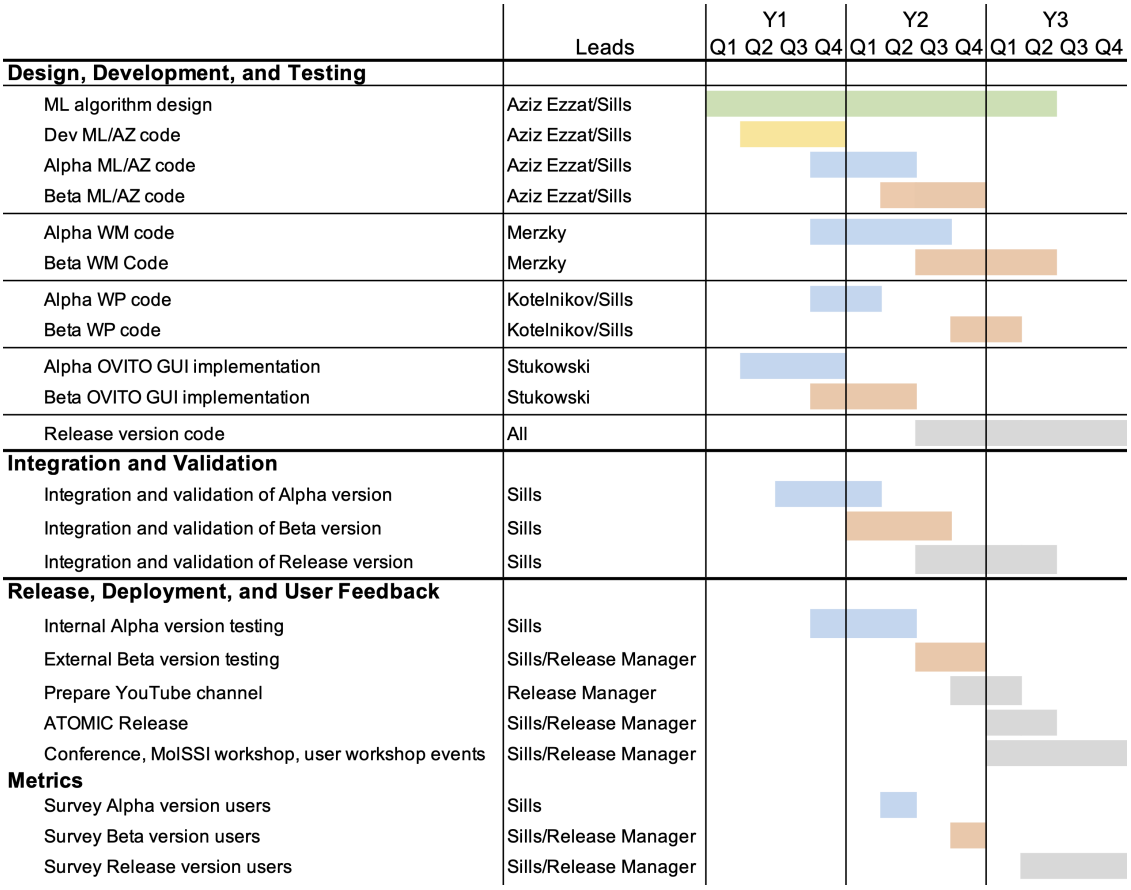


Figure 9: Gantt chart for the project.

$\text{ATOMIC}_{\text{AZ}}$ by users and (2) integration of $\text{ATOMIC}_{\text{ML}}$ with $\text{ATOMIC}_{\text{WM}}$ during class training and testing. For Integration Task 1, ATOMIC team users will search for and download class files from $\text{ATOMIC}_{\text{WP}}$, load them into $\text{ATOMIC}_{\text{AZ}}$, and analyze test simulaton datasets. ATOMIC team users will perform this integration testing using the OVITO-ATOMIC interface on various platforms (e.g., PC, Mac, Linux). For Integration Task 2, $\text{ATOMIC}_{\text{ML}}$ will be applied to several standard training problems. In terms of the compute resources managed by $\text{ATOMIC}_{\text{WM}}$ during integration testing, this is accomplished in two stages. First, $\text{ATOMIC}_{\text{WM}}$ will manage a pool of local HPC resources at Rutgers. In the second stage, $\text{ATOMIC}_{\text{WM}}$ will leverage distributed resources of various architectures from NSF HPC clusters accessed via an XSEDE Research Allocation.

For validation, an automated regression test library will be established. For new structure classes added to ATOMIC, a regression test will be established for each ATOMIC software package. These tests will ensure consistent performance and accuracy of ATOMIC, and safeguard against accidental deleterious modifications to source code. Regression tests will also be established at the code module level as necessary to validate accuracy of individual software modules.

2.1.6 Release, Deployment, and User Feedback

The release plan for ATOMIC will involve many facets and stages, as depicted in the Gantt chart (Fig. 9). Users will evaluate the performance and ease of use of ATOMIC via the OVITO-ATOMIC interface. ATOMIC will be released in three stages. First, Alpha version testing will be performed by internal ATOMIC team users, including high school interns and undergraduate researchers. Beta version testing will then be performed by a select group of colleagues in the materials science and atomistic modeling community, but external to the ATOMIC team.

At the end of Year 2, ATOMIC will be released to the public. ATOMIC will be released as part of OVITO Basic and also via a *public GitHub repository*, which will be announced via the ATOMIC and OVITO websites. To advertise ATOMIC and its capabilities to the community, a number of mechanisms will be employed. First, ATOMIC will be presented at several targeted technical conferences (e.g., TMS, MRS, MS&T) by organizing related symposia, holding information sessions, and renting exhibition booth space, the costs of which are included in the project budget. Secondly, we will create an ATOMIC YouTube channel where we will post video tutorials demonstrating how to use ATOMIC in OVITO, and how to interact with $\text{ATOMIC}_{\text{WP}}$. These videos will be embedded in the ATOMIC website in the Tutorials tab. Thirdly, ATOMIC will be presented and discussed at MolSSI and LAMMPS workshops. Fourthly, the ATOMIC team will host a virtual, invitation-only ATOMIC user workshop aimed at graduate students and postdocs, with the goal of advertising ATOMIC to the community and acquiring real-time feedback. At the workshop, users will be asked to perform several analyses in real-time given prepared atomistic datasets, after which they will provide feedback on their experience. They will also be asked to provide a test case of their own to evaluate. Fifthly, a journal article will be published which details the inner workings of ATOMIC, similar to a recent article by the OpenKIM team [65]. And finally, we will collect survey results from users at each stage of release (see Section 3.3).

The aggressive release plan detailed above requires coordination and organization of numerous events, web resources, and content creation. To that end, we will hire a senior undergraduate student to serve as a *Release Manager* for the project. This role serves to both relieve the PIs of some of the administrative and organizational duties while providing valuable, impactful experience for an undergraduate student in the release and evaluation of cyberinfrastructure (CI).

2.2 Building on existing, recognized capabilities

ATOMIC will heavily leverage existing NSF and other nationally funded resources, including:

- ATOMIC will draw on the expertise, knowledge-base, and resources of the *Molecular Sciences Software Institute (MolSSI)* (see letter of collaboration), an NSF-funded institute whose mission is to serve the worldwide community of computational molecular scientists. Specifically, we will: 1) consult with MolSSI on best practices for molecular sciences software development, 2) propose a MolSSI-sponsored workshop on software for atomistic structure analysis, and 3) apply to MolSSI's fellowship and summer school programs to educate the graduate students and postdocs involved with ATOMIC, and 4) advertise openings for postdoc and staff positions through MolSSI to attract strong team members. Our workshop proposal to MolSSI will be focused on pulling together experts across the many areas of science using atomistic simulations to work towards *a coherent and holistic view of the structure analysis needs of the community.*
- ATOMIC will leverage the NSF-funded *OpenKIM* program (see letter of collaboration), which provides a standardized software framework for storing, validating, and efficiently utilizing interatomic potentials for MD simulations. Using OpenKIM in our training/testing MD simulations, automatically documents and ensures the integrity and reproducibility of the utilized interatomic potentials (which govern interactions between the atoms).
- ATOMIC will utilize the NSF-funded *RADICAL Cybertools*, an abstractions-based suite of software modules to support scalable, interoperable and sustainable science on a range of high-performance and distributed computing infrastructure, in addition to other software from the Rutgers RADICAL group. RADICAL Cybertools will provide the backbone for $\text{ATOMIC}_{\text{WM}}$.
- ATOMIC will leverage the DoD-funded *AFLOW* program [66], a multifaceted consortium applying machine learning to materials discovery. In particular, the AFLOW Encyclopedia of Crystallographic Prototypes [20, 67, 68] provides a comprehensive digital library of perfect crystal structures. Within ATOMIC, we will use AFLOW's Prototype Label to uniquely identify the crystal structure for each structure class in a documented, reproducible manner.

- During development of ATOMIC's APIs, we will collaborate with the *SGX3 Center of Excellence for Science Gateways* (see letter of collaboration), an NSF-funded center which facilitates the sharing of experiences, technologies, and practices of those working with science gateways.
- MD simulations will utilized the efficient, massively parallel, and open-source *LAMMPS* software package, developed and supported by DOE.

2.3 Close collaborations among stakeholders

Our team integrates all key areas of expertise necessary for success in the development of ATOMIC: domain expertise in atomistic modeling and atomistic structures in metals (Sills [30, 69, 70], Stukowski [35, 71, 72]), graphical depiction of atomistic structures and user interfaces (Stukowski [29]), machine learning expertise for physical and materials sciences (Aziz Ezzat [73–75]), HPC and web server administration expertise (Kotelnikov [76]), and software engineering and HPC-at-scale expertise (Merzky [61, 77]). PI Sills will lead and manage the project as a whole, and leads for individual software development efforts are indicated in Fig. 9. Team and sub-team meetings will be held as necessary based on ongoing development and integration activities. For a discussion of the collaboration with OVITO GmbH, see below.

Engagement and collaboration with the broader atomistic modeling community will be achieved through the many events and activities discussed above, including: MolSSI workshop proposal on atomic structure analysis, ATOMIC invitation-only user workshop, ATOMIC beta testing by colleagues and collaborators, surveys and user feedback prompts in OVITO, and symposia, information sessions, and exhibit booths at conferences.

2.3.1 Collaboration with OVITO GmbH

Our collaboration with OVITO GmbH is a critical aspect of the project which ensures that ATOMIC is effective, easy to use, and widely available to the community. The OVITO visualization software is among the most widely used in the world. Integrating ATOMIC into OVITO gives us immediate access to thousands of existing and future OVITO users. Directly collaborating with OVITO GmbH for this integration ensures that the OVITO implementation is consistent with the goals and objectives of ATOMIC, while providing feedback to ATOMIC on how to optimize the user experience in OVITO. In addition, working with OVITO GmbH enables additional software features that would not be possible otherwise:

- *Direct integration between OVITO and ATOMIC$_{WP}$* – The OVITO interface will be setup so that the ATOMIC web portal can be directly queried and accessed *within* OVITO. This means users never have to leave OVITO to exercise the full capability of ATOMIC.
- *In-app user feedback prompts* – Feedback prompts will be integrated into OVITO while users are using ATOMIC, making it easy for users to provide feedback about their experience while it is fresh in their minds.
- *OVITO browser-based viewer in ATOMIC$_{WP}$* – Leveraging advancements at OVITO GmbH to integrate OVITO with analysis capabilities into a web browser, users will be able to visualize and manipulate structures while browsing/searching the structure database in ATOMIC$_{WP}$.

The Rutgers and OVITO GmbH teams will have periodic virtual meetings. PI Sills (Rutgers) and co-PI Stukowski (OVITO) have a track record of collaboration in software development [70, 78].

3 Measurable Outcomes

3.1 Deliverables

Each of the four software packages detailed in Section 2 will be available to the community via the ATOMIC website and the project's GitHub page. Additionally, ATOMIC$_{AZ}$ will be delivered to users as a standard feature in the OVITO visualization tool. All codes will be listed in the MolSSI Molecular Sciences Software Database to enable discovery by new users. The ATOMIC website will detail how to access and use the software.

3.2 Sustained and sustainable impacts

ATOMIC is designed so that its impact will grow over time as more and more structure classes are added to its database. In principle, the extent and duration of this impact is unbounded as the community of users and expanse of problems it is applied to grows. As discussed above, there is a surfeit of important atomic structures for which no analysis tools exist (e.g., Fig. 2), demonstrated an unmet need in the community which ATOMIC will fill.

We have designed ATOMIC to be a sustainable software platform in a number of ways. First, OVITO GmbH will manage and distribute $\text{ATOMIC}_{\text{AZ}}$ to users indefinitely at no cost. Secondly, web hosting and server needs for ATOMIC will be provided by Rutgers Engineering Computing Services indefinitely at no cost. Thirdly, and most importantly, ATOMIC is designed as a *community-based* tool. Widespread adoption and buy-in from the community will serve as a backstop for ATOMIC which sustains its availability and impact. This will manifest in several ways. For one, the compute resources necessary for training of new classes within $\text{ATOMIC}_{\text{ML}}$ will be provided on a volunteer basis by users and supporting institutions. $\text{ATOMIC}_{\text{WM}}$ is explicitly designed to manage and allocate work among this diverse, distributed set of computing resources. In addition, we anticipate that multiple instances of $\text{ATOMIC}_{\text{ML}}$ will be stood up by groups of stakeholders in different communities so that the specific needs of that community can be met most fully. In other words, both the coordination of computational tasks via $\text{ATOMIC}_{\text{ML}}$ and execution of these tasks via $\text{ATOMIC}_{\text{WM}}$ will be community-driven.

In terms of software documentation, as part of $\text{ATOMIC}_{\text{WP}}$ we will write and publish wiki-style documentation for $\text{ATOMIC}_{\text{AZ}}$, $\text{ATOMIC}_{\text{ML}}$, and $\text{ATOMIC}_{\text{WM}}$. This will include discussions of basic purpose and capabilities, examples and test cases, and an API reference for the software modules. Documentation for the OVITO-ATOMIC interface will be published on the OVITO web page [79] in the same style as other OVITO documentation.

3.3 Metrics

Since user feedback is the most straightforward way to assess the performance and impact of software (especially on the timescale of a three-year project), we will utilize user surveys as our primary quantitative metric. These surveys will include questions that are scored by the surveyee, so the quantitative results can be aggregated, and short-answer questions to allow for more specific feedback. Over the course of the project, surveys will be conducted against different user populations with the following timeline (see Fig. 9):

Year 1.5: Survey results from Alpha version users

Year 2: Survey results from Beta version users

Year 2.5+: Survey/OVITO feedback prompt results from Release version users

In addition to surveys, we will also be able to track the number of ATOMIC users by keeping track of the number of times users have accessed the structure class database on $\text{ATOMIC}_{\text{WP}}$.

4 Intellectual Merit

The intellectual merit of this project rests on three major innovations: (1) crowd-sourcing of automated, data-driven community-centered tools, (2) flexible and extensible analysis of atomic datasets, and (3) community-based HPC computing for CI. (1) ATOMIC will be a trailblazer for a new paradigm wherein data-driven, crowd-sourced approaches are adopted that enable automated subroutines which extend the capability of a tool given a new user request. Such a paradigm is broadly applicable across CI and beyond, and could be used to develop/advance tools in microscopy, spectroscopy, and analysis of other simulation datasets (e.g., field quantities, discrete elements). The $\text{ATOMIC}_{\text{ML}}$ software package will be the heart of this new paradigm by automating feature selection and classifier design within a scalable, adaptive framework. (2) The ATOMIC tool itself will revolutionize analysis of atomistic datasets by providing the first broadly

extensible structure analysis framework with unprecedented capability in terms of the ability to identify structures which no other technique can identify (e.g., vacancies, see Fig. 7). This capability derives from the use of machine learning to construct classifiers for each structure class, rather than ad hoc, heuristic rules [21–23, 35, 36]. (3) Finally, the ATOMIC$_{WM}$ software package will be a generic tool for managing large-scale computational workflows across diverse and varied computing resources, a valuable capability for many research areas. This will enable a new movement of community-based computing for HPC where the community contributes compute resources, an especially relevant innovation as the influence and need for ML continues expanding.

5 Broader Impacts

STEM outreach – The ATOMIC team will engage with local high school students through paid internships which will run for 8 weeks in Years 2 and 3 of this project. Participants will be be given support funds, which will enable underrepresented youth—who may not be able to participate otherwise—to apply to the program. To identify participants, the PIs will work with local community outreach groups including the YMCA of MEWSA and the Edison Municipal Youth Services Commission. Interns will be involved in: (1) software stress and performance testing, (2) Alpha/Beta testing, (3) setup and execution of regression test libraries, and (4) evaluation of ATOMIC's documentation, manuals, and examples/video tutorials. Interns will also get a glimpse into the industrial side of CI thanks to our GOALI partnership with OVITO GmbH.

Educational impact – PI Sills will incorporate usage of ATOMIC into his course "Multiscale Modeling of Materials," which is under development for the Materials Science and Engineering department. Co-PI Ezzat will integrate the ML aspects associated with this project as a case study in a senior undergraduate course on "Industrial Informatics" (ISE 540:485). Both PIs will identify several "side projects" as Senior capstone projects within their departments. They will also utilize undergraduate research programs at Rutgers University such as Aresty summer science and research assistant programs to engage undergraduate researchers.

Industrial and scientific communities – This project provides a unique opportunity to bridge the materials science, machine learning, and software engineering communities. This collaboration will serve as an exemplar to integration of machine learning and domain knowledge into CI design and development which will inspire similar future CI. Using ATOMIC, MD simulations will be able to address scientific and industrial research questions which are currently out of reach because the relevant atomic structures are impossible to analyze. The project will promote a community of researchers focused on advancing atomic structure analysis through organization of relevant conference symposia, MolSSI workshops, and through the ATOMIC$_{WP}$ forum and discussion board. The project will provide software engineering training to graduate students and postdocs through the MolSSI summer school, MolSSI-funded workshop, and collaboration with industrial partner OVITO GmbH. It will also provide coding and software development exposure for the next generation of engineers and scientists via our high school internship program.

6 Results from Prior NSF Support

Co-PIs Merzky and Stukowski have no NSF awards within 5 years of submission of this proposal. PI Sills and co-PI Aziz Ezzat serve as PI and co-PI, respectively, on Award #2034074, *Scale Bridging in Ductile Fracture via Kernel-based Machine Learning*, $576,943, 01/01/2022-12/31/2024; *Intellectual Merits:* To develop an atomistically informed continuum model for predicting ductile fracture in metals; *Broader Impacts:* Establishment of a new scale bridging framework for multiscale modeling and introduction of atomistic modeling and ML in a high school classroom. The project has resulted so far in one published article [80] and one article under review.

E - REFERENCES CITED

[1] Daan Frenkel and Berend Smit. *Understanding Molecular Simulation: From Algorithms to Applications*. Number 1 in Computational Science Series. Academic Press, San Diego, 2nd ed edition, 2002.

[2] David S. Sholl and Janice A. Steckel. *Density Functional Theory: A Practical Introduction*. Wiley, Hoboken, N.J, 2009.

[3] W. Cai, J. Li, and S. Yip. Molecular Dynamics. In *Comprehensive Nuclear Materials*, pages 249–265. Elsevier, 2012.

[4] Yang Yang, Harith Humadi, Dorel Buta, Brian B. Laird, Deyan Sun, Jeffrey J. Hoyt, and Mark Asta. Atomistic simulations of nonequilibrium crystal-growth kinetics from alloy melts. *Phys. Rev. Lett.*, 107:025505, Jul 2011.

[5] Markus J. Buehler. *Atomistic Modeling of Materials Failure*. Springer, New York, N.Y, 2010.

[6] P Andric and W A Curtin. Atomistic modeling of fracture. *Modelling and Simulation in Materials Science and Engineering*, 27(1):013001, January 2019.

[7] Tengfei Luo and Gang Chen. Nanoscale heat transfer – from computation to experiment. *Physical Chemistry Chemical Physics*, 15(10):3389, 2013.

[8] D. J Fisher. *Molecular Dynamics and Diffusion: A Compilation*. 2013.

[9] Ferdinand Evers, Richard Korytár, Sumit Tewari, and Jan M van Ruitenbeek. Advances and challenges in single-molecule electron transport. *Reviews of Modern Physics*, 92(3):035001, 2020.

[10] Marius Bürkle, Umesha Perera, Florian Gimbert, Hisao Nakamura, Masaaki Kawata, and Yoshihiro Asai. Deep-learning approach to first-principles transport simulations. *Phys. Rev. Lett.*, 126:177701, Apr 2021.

[11] Jun Zhong, Donald J. Siegel, Louis G. Hector Jr., and James B. Adams. *Atomistic Simulations of Adhesion, Indentation and Wear at the Nanoscale*, chapter 25, pages 601–645. John Wiley Sons, Ltd, 2017.

[12] Robert B Best. Atomistic molecular simulations of protein folding. *Current Opinion in Structural Biology*, 22(1):52–61, 2012. Folding and binding/Protein-nucleic acid interactions.

[13] Gerhard Goldbeck. The economic impact of molecular modelling. Technical report, Goldbeck Consulting, 2012.

[14] `https://www.lammps.org/download.html`, accessed November 2021.

[15] A.M. Cuitiño and M. Ortiz. Ductile fracture by vacancy condensation in f.c.c. single crystals. *Acta Materialia*, 44(2):427–436, 1996.

[16] M.E. Kassner and T.A. Hayes. Creep cavitation in metals. *International Journal of Plasticity*, 19(10):1715–1748, 2003.

[17] Michihiko Nagumo and Kenichi Takai. The predominant role of strain-induced vacancies in hydrogen embrittlement of steels: Overview. *Acta Materialia*, 165:722–733, 2019.

[18] M.S. Veshchunov. On the theory of void nucleation in irradiated crystals. *Journal of Nuclear Materials*, 571:154021, 2022.

[19] S. Saimoto and B.J. Diak. Point defect generation, nano-void formation and growth. i. validation. *Philosophical Magazine*, 92(15):1890–1914, 2012.

[20] AFLOW Encyclopedia of Crystallographic Prototypes. `http://aflowlib.org/prototype-encyclopedia/`.

[21] Peter Mahler Larsen, Søren Schmidt, and Jakob Schiøtz. Robust structural identification via polyhedral template matching. *Modelling and Simulation in Materials Science and Engineering*, 24(5):055007, June 2016.

[22] Andrew H. Nguyen and Valeria Molinero. Identification of Clathrate Hydrates, Hexagonal Ice, Cubic Ice, and Liquid Water in Simulations: The CHILL+ Algorithm. *The Journal of Physical Chemistry B*, 119(29):9369–9376, July 2015.

[23] Lei Deng, Xingming Zhang, Liang Wang, Jianfeng Tang, Zhixiao Liu, Shifang Xiao, Huiqiu Deng, and Wangyu Hu. Local identification of chemical ordering: Extension, implementation, and application of the common neighbor analysis for binary systems. *Computational Materials Science*, 143:195–205, February 2018.

[24] Teppei Fukuya and Yasushi Shibuta. Machine learning approach to automated analysis of atomic configuration of molecular dynamics simulation. *Computational Materials Science*, 184:109880, November 2020.

[25] Heejung W. Chung, Rodrigo Freitas, Gowoon Cheon, and Evan J. Reed. Data-centric framework for crystal structure identification in atomistic simulations using machine learning. *Physical Review Materials*, 6(4):043801, April 2022.

[26] Open Knowledge of Interatomic Models. `https://openkim.org/`.

[27] Folding@Home. `https://foldingathome.org/?lng=en`.

[28] QCArchive. `https://qcarchive.molssi.org/`.

[29] Alexander Stukowski. Visualization and analysis of atomistic simulation data with OVITO–the Open Visualization Tool. *Modelling and Simulation in Materials Science and Engineering*, 18(1):015012, January 2010.

[30] R. B. Sills and B. L. Boyce. Void growth by dislocation adsorption. *Materials Research Letters*, 8(3):103–109, March 2020.

[31] Mirella S. Santos, LuÃs F. M. Franco, Marcelo Castier, and Ioannis G. Economou. Molecular dynamics simulation of n-alkanes and co2 confined by calcite nanopores. *Energy & Fuels*, 32(2):1934–1941, 2018.

[32] Guangji Xu and Hao Wang. Study of cohesion and adhesion properties of asphalt concrete with molecular dynamics simulation. *Computational Materials Science*, 112:161–169, 2016.

[33] Jonathan D. Halverson, Won Bo Lee, Gary S. Grest, Alexander Y. Grosberg, and Kurt Kremer. Molecular dynamics simulation study of nonconcatenated ring polymers in a melt. ii. dynamics. *The Journal of Chemical Physics*, 134(20):204905, 2011.

[34] Ashivni Shekhawat and Robert O. Ritchie. Toughness and strength of nanocrystalline graphene. *Nature Communications*, 7(1):10546, April 2016.

[35] Alexander Stukowski. Structure identification methods for atomistic simulations of crystalline materials. *Modelling and Simulation in Materials Science and Engineering*, 20(4):045021, June 2012.

[36] Emanuel A Lazar. *VoroTop* : Voronoi cell topology visualization and analysis toolkit. *Modelling and Simulation in Materials Science and Engineering*, 26(1):015011, January 2018.

[37] Aidan P. Thompson, H. Metin Aktulga, Richard Berger, Dan S. Bolintineanu, W. Michael Brown, Paul S. Crozier, Pieter J. in 't Veld, Axel Kohlmeyer, Stan G. Moore, Trung Dac Nguyen, Ray Shan, Mark J. Stevens, Julien Tranchida, Christian Trott, and Steven J. Plimpton. LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Computer Physics Communications*, 271:108171, February 2022.

[38] Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS). `https://www.lammps.org/`.

[39] TensorFlow. `https://www.tensorflow.org/`.

[40] Voro++. `http://math.lbl.gov/voro++/`.

[41] Cenk Kaynak and Ethem Alpaydin. Multistage cascading of multiple classifiers: One man's noise is another man's data. In *ICML*, pages 455–462. Citeseer, 2000.

[42] Lester Mackey, Jordan Bryan, and Man Yue Mo. Weighted classification cascades for optimizing discovery significance in the higgsml challenge. In *NIPS 2014 Workshop on High-energy Physics and Machine Learning*, pages 129–134. PMLR, 2015.

[43] Lubomir Bourdev and Jonathan Brandt. Robust object detection via soft cascade. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 236–243. IEEE, 2005.

[44] Mohammad J Saberian and Nuno Vasconcelos. Boosting classifier cascades. In *NIPS*, volume 23, pages 2047–2055. Citeseer, 2010.

[45] Sumit Chopra, Suhrid Balakrishnan, and Raghuraman Gopalan. Dlid: Deep learning for domain adaptation by interpolating between domains. In *ICML workshop on challenges in representation learning*, volume 2. Citeseer, 2013.

[46] Alon Zweig and Daphna Weinshall. Hierarchical regularization cascade for joint learning. In *International Conference on Machine Learning*, pages 37–45. PMLR, 2013.

[47] Lauri Himanen, Marc O.J. Jäger, Eiaki V. Morooka, Filippo Federici Canova, Yashasvi S. Ranawat, David Z. Gao, Patrick Rinke, and Adam S. Foster. DScribe: Library of descriptors for machine learning in materials science. *Computer Physics Communications*, 247:106949, February 2020.

[48] Conrad W. Rosenbrock, Eric R. Homer, Gábor Csányi, and Gus L. W. Hart. Discovering the building blocks of atomic systems using machine learning: Application to grain boundaries. *npj Computational Materials*, 3(1):29, December 2017.

[49] Rohit Batra, Huan Doan Tran, Chiho Kim, James Chapman, Lihua Chen, Anand Chandrasekaran, and Rampi Ramprasad. General Atomic Neighborhood Fingerprint for Machine Learning-Based Methods. *The Journal of Physical Chemistry C*, 123(25):15859–15866, June 2019.

[50] Amit Samanta. Representing local atomic environment using descriptors based on local correlations. *The Journal of Chemical Physics*, 149(24):244102, December 2018.

[51] Ralf Drautz. Atomic cluster expansion for accurate and transferable interatomic potentials. *Physical Review B*, 99(1):014104, January 2019.

[52] Ralf Drautz. Atomic cluster expansion of scalar, vectorial, and tensorial properties including magnetism and charge transfer. *Physical Review B*, 102(2):024104, July 2020.

[53] Wesley F. Reinhart and Athanassios Z. Panagiotopoulos. Automated crystal characterization with a fast neighborhood graph analysis method. *Soft Matter*, 14(29):6083–6089, 2018.

[54] Brandon D. Snow, Dustin D. Doty, and Oliver K. Johnson. A Simple Approach to Atomic Structure Characterization for Machine Learning of Grain Boundary Structure-Property Models. *Frontiers in Materials*, 6:120, May 2019.

[55] Emanuel A. Lazar, Jian Han, and David J. Srolovitz. Topological framework for local structure analysis in condensed matter. *Proceedings of the National Academy of Sciences*, 112(43):E5769–E5776, October 2015.

[56] Arash Dehghan Banadaki and Srikanth Patala. A three-dimensional polyhedral unit model for grain boundary structure in fcc metals. *npj Computational Materials*, 3(1):13, December 2017.

[57] Jiliang Tang, Salem Alelyani, and Huan Liu. Feature selection for classification: A review. *Data classification: Algorithms and applications*, page 37, 2014.

[58] Michael J Swain and Dana H Ballard. Color indexing. *International journal of computer vision*, 7(1):11–32, 1991.

[59] Matteo Turilli, Vivek Balasubramanian, Andre Merzky, Ioannis Paraskevakos, and Shantenu Jha. Middleware building blocks for workflow systems. *Computing in Science & Engineering*, 21(4):62–75, 2019.

[60] Vivek Balasubramanian, Matteo Turilli, Weiming Hu, Matthieu Lefebvre, Wenjie Lei, Ryan Modrak, Guido Cervone, Jeroen Tromp, and Shantenu Jha. Harnessing the power of many: Extensible toolkit for scalable ensemble applications. In *International Parallel and Distributed Processing Symposium*, pages 536–545. IEEE, 2018.

[61] Andre Merzky, Matteo Turilli, Manuel Maldonado, Mark Santcroos, and Shantenu Jha. Using pilot systems to execute many task workloads on supercomputers. In *Workshop on Job Scheduling Strategies for Parallel Processing*, pages 61–82. Springer, 2018.

[62] Andre Merzky, Ole Weidner, and Shantenu Jha. SAGA: A standardized access layer to heterogeneous distributed computing infrastructure. *Software-X*, 2015. DOI: 10.1016/j.softx.2015.03.001.

[63] Matteo Turilli, Mark Santcroos, and Shantenu Jha. A comprehensive perspective on pilot-job systems. *ACM Comput. Surv.*, 51(2):43:1–43:32, April 2018.

[64] Hyungro Lee, Matteo Turilli, Shantenu Jha, Debsindhu Bhowmik, Heng Ma, and Arvind Ramanathan. Deepdrivemd: Deep-learning driven adaptive molecular simulations for protein folding. In *2019 IEEE/ACM Third Workshop on Deep Learning on Supercomputers (DLS)*, pages 12–19. IEEE, 2019.

[65] D. S. Karls, M. Bierbaum, A. A. Alemi, R. S. Elliott, J. P. Sethna, and E. B. Tadmor. The OpenKIM processing pipeline: A cloud-based automatic material property computation engine. *The Journal of Chemical Physics*, 153(6):064104, August 2020.

[66] AFLOW: Automatic FLOW for Materials Discovery. `http://aflowlib.org/`.

[67] Michael J. Mehl, David Hicks, Cormac Toher, Ohad Levy, Robert M. Hanson, Gus Hart, and Stefano Curtarolo. The AFLOW Library of Crystallographic Prototypes: Part 1. *Computational Materials Science*, 136:S1–S828, August 2017.

[68] David Hicks, Michael J. Mehl, Eric Gossett, Cormac Toher, Ohad Levy, Robert M. Hanson, Gus Hart, and Stefano Curtarolo. The AFLOW Library of Crystallographic Prototypes: Part 2. *Computational Materials Science*, 161:S1–S1011, April 2019.

[69] Ryan B. Sills, Michael E. Foster, and Xiaowang W. Zhou. Line-length-dependent dislocation mobilities in an FCC stainless steel alloy. *International Journal of Plasticity*, 135:102791, December 2020.

[70] Nipal Deka, Alexander Stukowski, and Ryan B. Sills. Automated extraction of interfacial dislocations and disconnections from atomistic data. *Acta Materialia*, 256:119096, September 2023.

[71] Alexander Stukowski and Karsten Albe. Extracting dislocations and non-dislocation crystal defects from atomistic simulation data. *Modelling and Simulation in Materials Science and Engineering*, 18(8):085001, December 2010.

[72] Alexander Stukowski. Dislocation Analysis Tool for Atomistic Simulations. In Wanda Andreoni and Sidney Yip, editors, *Handbook of Materials Modeling*, pages 1545–1558. Springer International Publishing, Cham, 2020.

[73] Ahmed Aziz Ezzat and Mostafa Bedewy. Machine learning for revealing spatial dependence among nanoparticles: Understanding catalyst film dewetting via gibbs point process models. *The Journal of Physical Chemistry C*, 124(50):27479–27494, 2020.

[74] Ahmed Aziz Ezzat, Arash Pourhabib, and Yu Ding. Sequential design for functional calibration of computer models. *Technometrics*, 60(3):286–296, 2018.

[75] Ahmed Aziz Ezzat, Sheng Liu, Dorit S Hochbaum, and Yu Ding. A graph-theoretic approach for spatial filtering and its impact on mixed-type spatial pattern recognition in wafer bin maps. *IEEE Transactions on Semiconductor Manufacturing*, 34(2):194–206, 2021.

[76] Computing Environment Course taught in the Rutgers School of Engineering. `http://coewww.rutgers.edu/www1/linuxclass2021/lessons/html/index.html`.

[77] Matteo Turilli, Andre Merzky, Vivek Balasubramanian, and Shantenu Jha. Building blocks for workflow system middleware. In *18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGRID 2018, Washington, DC, USA, May 1-4, 2018*, pages 348–349. IEEE, 2018.

[78] ILDA: Interfacial Line Defect Analysis tool. `https://gitlab.com/mmod_public/ilda`.

[79] OVITO: Open Visualization Tool. `https://www.ovito.org/`.

[80] Philip J. Noell, Ryan B. Sills, Ahmed Amine Benzerga, and Brad L. Boyce. Void nucleation during ductile rupture of metals: A review. *Progress in Materials Science*, 135:101085, June 2023.

FACILITIES, EQUIPMENT, AND OTHER RESOURCES

The Rutgers School of Engineering will provide office space for both the PIs (excluding external co-PI Stukowski) and the graduate students/postdocs funded under this project. The PI (Sills) has a 180 sq. ft. private office in the Center for Ceramics Research on the Busch Campus where the School of Engineering is located. The PI's research group has a 540 sq. ft. office in the Engineering A-Wing adjacent to the Center for Ceramics Research with 6 desks and a large meeting table where his postdoc will have a desk. Co-PI (Aziz Ezzat) has a 150 sq. ft. private office in the Computing Research & Education (CoRE) Building on the Busch Campus where the School of Engineering is located. Aziz Ezzat's research group has a three-room lab in the Richard Weeks Hall of Engineering, shared with two ISE faculty. The lab hosts Dr. Aziz Ezzat research group's `DELL PowerEdge T640` research server with 2TB of storage capacity and 2 `Intel Xeon Gold 6132` processors for data processing capability. His graduate student under this project will be given a desk in this lab space.

Computational Resources

PI Sills has a Macbook Pro purchased in 2019 as his primary computer. Co-PI Aziz Ezzay has an ASUS Q536FD notebook purchased in 2019 as his primary computer. The Rutgers School of Engineering (SoE) provides research faculty and team needs with robust high-performance computing resources, supported by dedicated technical staff. Simulations and computations under this project will in part leverage the SoE HPC cluster with $\sim$ 3000 compute cores and $\sim$ 10 compute GPUs. No funds from this project will go towards maintenance of the cluster. The SoE cluster features FDR 56 Gbit infiniband, Gbit networks, a distributed BeeGFS cluster file system for parallel multi-node simulations, and a SLURM queuing system. The NFS file system for long-term data storage provides the same file system image to all nodes of the cluster. The following applications of relevance to the project are installed on the SoE cluster: Intel C/C++/Fortran 2018 compilers, GNU GCC/G++/gfortran 5.4 compilers, `Open MPI` (compiled with GNU and Intel compilers), `LAMMPS`, , `R`, `MATLAB`, and `CUDA`.

One aspect of the project is the development of a platform agnostic computation management platform, ATOMIC$_{\text{WM}}$. An important aspect of its development will be performance evaluation using a variety of compute resource architectures. This will be accomplished using two mechanisms: (1) using local compute resources at Rutgers and (2) using NSF compute resources, to be accessed via an XSEDE Research Allocation which will be pursued after the project begins.

Other Support Staff

Rutgers University will provide support staff to aid with project administration and financials. Alexei Kotelnikov, Associate Director of Information Technology in SoE, and other Engineering Computing Services staff will support the project by managing, configuring, and maintaining Rutgers compute resources and servers used for ATOMIC.

Incomputable Solutions Inc. Facilities

Incomputable Solutions Inc. is a registered small business that delivers software solutions to execute and manage end-to-end computational campaigns on scalable computing platforms. It uses advanced cloud computing services for its computing, data storage, publishing, and financials. It has servers for local on-premise data backups. Incomputable Solutions Inc. also has dedicated office space in New Brunswick adjacent to Rutgers University.

Other Resources

Industrial co-PI Dr. Alexander Stukowski will support development of the ATOMIC-OVITO interface using computing systems at OVITO GmbH. These systems are designed for testing

OVITO's performance on a range of platforms and operating systems (e.g., PC and Mac). Dr. T. Daniel Crawford and MolSSI will provide general software development support. Dr. Michael Zentner will support development of the web portal and workflow manager. Dr. Ellad Tadmor will support integration of ATOMIC with OpenKIM.

Data Management Plan

1. Types of Data

The data generated in this project will principally be comprised of source code and script files which make up the software packages, scripts used to analyze and evaluate results, simulation data files, and text documents. These will mostly take the form of ASCII text files written in the syntax/language relevant to the particular application (e.g., C++, LAMMPS). Text documents, in the form of reports, manuscripts, presentations, and manuals, will be written in Microsoft Office products or Latex.

2. Data Standards

Text files will be stored in standard ASCII format. The syntax for each file will be consistent with established standards for the respective language/application. Text documents will follow established .docx/.pptx and/or .tex standards.

3. Policies for Access and Sharing

All codes produced under this project will be open-source under the MIT license. All simulation results used to train and generate structure classifiers will be publicly available through the ATOMIC web portal, along with the performance metrics for the associated ATOMIC classifier. All text documents will be shared publicly as reports, manuscripts, or presentations through appropriate publication channels (e.g., journal submission, conference submission).

4. Re-use, Re-distribution, and Production of Derivatives

Re-use and production of derivatives from our source codes will be subject to the open-source MIT license under which they are released. Simulation data will be freely available for re-use, pending the ATOMIC project is cited appropriately upon their re-use.

5. Archiving and Preservation of Access

All source code will be revision controlled and archived on GitHub. All training and testing data will be archived on backed-up Rutgers file servers. Data used for publications will be archived and made available via Data DOIs using free data archiving services, such as Zenodo (zenodo.org).

Delivery Mechanism and Community Usage Metrics

1. Deliverables

The four ATOMIC software packages will be delivered via the following mechanisms:

- *GitHub* – All source codes developed under this project will be freely available via the ATOMIC GitHub page. This includes the $\text{ATOMIC}_{\text{ML}}$ and $\text{ATOMIC}_{\text{WM}}$ codes. While specialized for the problem at hand, the $\text{ATOMIC}_{\text{ML}}$ code will serve as an exemplar in applied machine learning which can be modified for other applications. $\text{ATOMIC}_{\text{WM}}$ will be developed as a generic tool for managing execution of compute tasks on diverse, distributed resources.

- *OVITO* – $\text{ATOMIC}_{\text{AZ}}$ will be bundled as a standard feature with the open-source OVITO Basic visualization tool. OVITO will also come pre-packaged with a few commonly utilized structure class files. From inside OVITO, users will be able to directly access $\text{ATOMIC}_{\text{WP}}$ in order to search and download class files and submit requests for new structures.

- $ATOMIC_{WP}$ – The ATOMIC web portal, $\text{ATOMIC}_{\text{WP}}$, will provide a delivery vessel for two key ATOMIC services. First, it will house all structure class files in a searchable database so that users can easily determine if a class file of interest exists. Secondly, users will be able to submit requests for new structure analysis capabilities through the web portal. The long-term vision is that this request will kick-off a series of automated computations that culminate in the creation of a new class file. However, for the purposes of the current project, these requests will simply be collected together and manually reviewed.

2. Metrics

The primary metric for the impact of ATOMIC will be survey results from users. These surveys are staged to align with release events, e.g., Alpha, Beta, and Release version surveys. Surveys will be comprised of a series of questions and statements requesting quantitative answers such as: "How satisfied are you with your experience using ATOMIC? (10 = very satisfied, 0 = very dissatisfied)" and "ATOMIC enabled me to analyze my simulations and unprecedented ways (10 = strongly, 0 = not at all)." These quantitative scores will be pooled together to provide an overall set of scores for each release stage. Comparisons will be made across release stages to ensure that each subsequent release is incorporating user feedback in a constructive way. Given the uncertainties regarding the details of these surveys, it is difficult to set precise quantitative yearly targets. Our goal will simply be to improve our survey scores with each subsequent release.

In addition to formal, comprehensive surveys, we will also incorporate in-app feedback prompts into OVITO so that users can provide feedback while they are using ATOMIC. This data will be collected on an ongoing basis after ATOMIC is released to the public. Again, it is difficult to set quantitative yearly targets, so the goal will be to pursue "increasing" feedback scores over time.

Finally, we will be able to roughly track usage of ATOMIC by tracking the number of times the web portal, $\text{ATOMIC}_{\text{WP}}$, is accessed. Users will access $\text{ATOMIC}_{\text{WP}}$ whenever they need a new structure class file or wish to request creation of a new structure class file. While this is not a direct measure of how often ATOMIC is used to analyze atomistic datasets (which is the ultimate measure of impact), it provides some information on ATOMIC's impact on the research community. Our goal is for the web portal to have 1000 hits per month by the end of the project.

CI Professional Mentoring and/or Professional Development Plan

In order to mentor and enable professional development of CI professionals involved in the project, we will leverage the many programs available at Rutgers. These programs include:

- *Career advancement and training* – Rutgers Career Exploration and Success offers a wide array of services to the Rutgers community including career counseling and planning, with services organized based on different classes of communities. They also offer a busy event calendar (typically 2-6 events per day) that is open to the entire Rutgers community. These events include workshops aimed at professional development, counseling and guidance on career advancement, information and guidance sessions with industrial guest speakers, and seminars on well-being and health. Representative examples include: "Less Stress and More Success: Strategies to Manage the Stress of Academics, Work, and the Job Search" and "VMware Professional Selling Workshop."

- *Diversity, Equity, and Inclusion* - Rutgers University Equity and Inclusion offers a range of services to educate and engage the Rutgers community on issues related to diversity, equity, and inclusion. Workshops are aimed at community and personal growth, with topics such as "Get Recognized: Strategies for Self-Promotion." Weekly webinars cover topics on communication, community building, and effective use of technology.

- *Writing skill advancement* – The Rutgers Office of the Executive Vice President for Academic Affairs offers Writing Support for faculty and staff. This includes "Writing Retreats" which feature guest speakers covering a broad range of topics in writing, such as getting published, overcoming writer's block, and developing writing routines. They also host "Writing Accountability Groups" to help staff and faculty schedule and prioritize writing.

All of these programs and more will be made available to CI professionals engaged in the project so that they can cater their development plan in the manner that best suites their needs and career stage.

In addition, we will leverage the many career and professional skills development opportunities for CI professionals at MolSSI. These include workshops on software, HPC, and ML, and industrial training led by MolSSI and industrial partners such as NVIDIA and Intel.

The PIs will take an active role in mentoring CI professionals on the project team. This will include: (1) an initial discussion of all of the university resources available to them and how to effectively utilize them, (2) bimonthly "check-ins" to discuss progress on and identify opportunities for career development, and (3) bimonthly engagement with CI professionals on the "bigger picture" of the project to enable development of project management skills.

Postdoctoral Researcher Mentoring Plan

In order to mentor and enable professional development of postdoctoral researchers involved in the project, we will leverage the many programs available at Rutgers. These programs include:

- *Postdoctoral advancement* – Rutgers Office of Postdoctoral Affairs offers a range of services and events aimed an advancing postdocs and promoting the community. Monthly activities and events are designed around career development and community building. The Office also makes a broad suite of tools available for career planning, mentoring, and instruction on grant writing.

- *Career advancement and training* – Rutgers Career Exploration and Success offers a wide array of services to the Rutgers community including career counseling and planning, with services organized based on different classes of communities. They also offer a busy event calendar (typically 2-6 events per day) that is open to the entire Rutgers community. These events include workshops aimed at professional development, counseling and guidance on career advancement, information and guidance sessions with industrial guest speakers, and seminars on well-being and health. Representative examples include: "Less Stress and More Success: Strategies to Manage the Stress of Academics, Work, and the Job Search" and "VMware Professional Selling Workshop."

- *Diversity, Equity, and Inclusion* - Rutgers University Equity and Inclusion offers a range of services to educate and engage the Rutgers community on issues related to diversity, equity, and inclusion. Workshops are aimed at community and personal growth, with topics such as "Get Recognized: Strategies for Self-Promotion." Weekly webinars cover topics on communication, community building, and effective use of technology.

- *Writing skill advancement* – The Rutgers Office of the Executive Vice President for Academic Affairs offers Writing Support for faculty and staff. This includes "Writing Retreats" which feature guest speakers covering a broad range of topics in writing, such as getting published, overcoming writer's block, and developing writing routines. They also host "Writing Accountability Groups" to help staff and faculty schedule and prioritize writing.

All of these programs and more will be made available to postdoctoral researchers. The PIs will work with the postdoctoral researchers to help them identify growth areas which would benefit most from additional development.

In addition, we will leverage the many career and professional skills development opportunities for postdocs at MolSSI. These include workshops on software, HPC, and ML, and fellowship opportunities including 1-year of support working with a MolSSI mentor.

The PIs will take an active role in mentoring postdoctoral researchers on the project team. This will include: (1) an initial discussion of all of the university resources available to the postdoc and how to effectively utilize them, (2) monthly one-on-one meetings with a PI to discuss professional progress and skill development/growth areas, (3) a one-on-one meeting with a PI 3-6 months prior to departure to discuss career next-steps and career planning with bimonthly follow ups to check on progress, and (4) engagement with the postdoc regarding the PI's ongoing proposal preparation and peer-review activities to help train them in these vital aspects of science. Finally, PIs will encourage postdocs to present at (at least) one technical conference during the project, and work with them to learn to effectively disseminate their research findings.

Project Personnel and Partner Organizations

1. Ryan Sills; Rutgers University; PI
2. Ahmed Aziz Ezzat; Rutgers University; co-PI
3. Andre Merzky; Incomputable Solutions LLC; co-PI and Subawardee
4. Alexander Stukowski; OVITO GmbH; Industrial co-PI
5. Alexei Kotelnikov; Rutgers University; Unpaid Collaborator
6. T. Daniel Crawford; The Molecular Sciences Software Institute; Unpaid Collaborator
7. Michael Zentner; SGX3 Center of Excellence for Science Gateways; Unpaid Collaborator
8. Ellad Tadmor; University of Minnesota; Unpaid Collaborator