IMPECCABLE: Integrated Modeling PipelinE for COVID Cure by Assessing Better LEads

Aymen Al Saadi
Rutgers University, New Brunswick

Dario Alfe
University College London and University of Naples Federico II

Yadu Babuji
University of Chicago

Agastya Bhati
University College London

Ben Blaiszik
University of Chicago and Argonne National Laboratory

Alexander Brace
Argonne National Laboratory

Thomas Brettin
Argonne National Laboratory

Kyle Chard
University of Chicago and Argonne National Laboratory

Ryan Chard
University of Chicago

Austin Clyde
University of Chicago and Argonne National Laboratory

Peter Coveney*
University College London and University of Amsterdam

Ian Foster
University of Chicago and Argonne National Laboratory

Tom Gibbs
NVIDIA Corporation

Shantenu Jha†
Brookhaven National Laboratory, and Rutgers University, New Brunswick

Kristopher Keipert
NVIDIA Corporation

Thorsten Kurth
NVIDIA Corporation

Dieter Kranzlmüller
Leibniz Supercomputing Centre

Hyungro Lee
Rutgers University, New Brunswick

Zhuozhao Li
University of Chicago

Heng Ma
Argonne National Laboratory

Andre Merzky
Rutgers University, New Brunswick

Gerald Mathias
Leibniz Supercomputing Centre

Alexander Partin
Argonne National Laboratory

Junqi Yin
Oak Ridge Leadership Computing Facility

Arvind Ramanathan‡
University of Chicago and Argonne National Laboratory

Ashka Shah
Argonne National Laboratory

Abraham Stern
NVIDIA Corporation

Rick Stevens§
University of Chicago and Argonne National Laboratory

Li Tan
Brookhaven National Laboratory

Mikhail Titov
Rutgers University, New Brunswick

Anda Trifan
Argonne National Laboratory

Aristeidis Tsaris
Oak Ridge Leadership Computing Facility

Matteo Turilli
Rutgers University, New Brunswick

Huub Van Dam
Brookhaven National Laboratory

Shunzhou Wan
University College London

David Wifling
Leibniz Supercomputing Centre

*contact author, p.v.coveney@ucl.ac.uk
†contact author, shantenu@bnl.gov
‡contact author, ramanathana@anl.gov
§contact author, stevens@anl.gov

ICPP '21, August 9–12, 2021, Lemont, IL, USA

ACM ISBN 978-1-4503-9068-2/21/08…$15.00
https://doi.org/10.1145/3472456.3473524

ABSTRACT

The drug discovery process currently employed in the pharmaceutical industry typically requires about 10 years and $2–3 billion to deliver one new drug. This is both too expensive and too slow, especially in emergencies like the COVID-19 pandemic. In silico methodologies need to be improved both to select *better* lead compounds, so as to improve the efficiency of later stages in the drug discovery protocol, and to identify those lead compounds *more quickly*. No known methodological approach can deliver this combination of higher quality *and* speed. Here, we describe an Integrated Modeling PipEline for COVID Cure by Assessing Better LEads (IMPECCABLE) that employs multiple methodological innovations to overcome this fundamental limitation. We also describe the computational framework that we have developed to support these innovations at scale, and characterize the performance of this framework in terms of throughput, peak performance, and scientific results. We show that individual workflow components deliver 100× to 1000× improvement over traditional methods, and that the integration of methods, supported by scalable infrastructure, speeds up drug discovery by orders of magnitudes. IMPECCABLE has screened $\sim 10^{11}$ ligands and has been used to discover a promising drug candidate. These capabilities have been used by the US DOE National Virtual Biotechnology Laboratory and the EU Centre of Excellence in Computational Biomedicine.

ACM Reference Format:
Aymen Al Saadi, Dario Alfe, Yadu Babuji, Agastya Bhati, Ben Blaiszik, Alexander Brace, Thomas Brettin, Kyle Chard, Ryan Chard, Austin Clyde, Peter Coveney, Ian Foster, Tom Gibbs, Shantenu Jha, Kristopher Keipert, Thorsten Kurth, Dieter Kranzlmüller, Hyungro Lee, Zhuozhao Li, Heng Ma, Andre Merzky, Gerald Mathias, Alexander Partin, Junqi Yin, Arvind Ramanathan, Ashka Shah, Abraham Stern, Rick Stevens, Li Tan, Mikhail Titov, Anda Trifan, Aristeidis Tsaris, Matteo Turilli, Huub Van Dam, Shunzhou Wan, and David Wifling. 2021. IMPECCABLE: Integrated Modeling PipelinE for COVID Cure by Assessing Better LEads. In *50th International Conference on Parallel Processing (ICPP '21), August 9–12, 2021, Lemont, IL, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3472456.3473524

1 INTRODUCTION

Drug discovery is an astonishingly resource intensive process; the average time to search, design, and effectively bring a clinically tested drug can range between 10 to 15 years, and can cost over $1B [24]. Early stages of drug discovery rely on high throughput screening (HTS) protocols to identify chemical compounds that can be effective against known protein targets [11]. As more accurate screening methods are typically also more expensive, traditional virtual HTS employs a multi-stage pipeline protocol [23], wherein downstream stages are computationally more expensive, but typically more accurate in their selection of promising candidates. The number of candidates screened in downstream stages of a virtual HTS pipeline is less than upstream stages.

Although HTS approaches are widely available and used, the sheer combinatorics of drug-like molecules—an estimated 10^{68} possible compounds—makes it infeasible to exhaustively examine compound space to find viable molecules [8]. Computational frameworks that support both high-throughput exploration and accurate prediction of drug-like properties are needed.

In the context of COVID-19, the SARS-CoV-2 genome consists of 29 proteins of which 16 non-structural proteins are enzymes that play critical roles in the virus life cycle [22]. Identifying compounds that have the potential to inhibit the virus life cycle [22] requires screening *tens of billions* of small molecules against multiple targets on the SARS-CoV-2 proteome. Thus, virtual HTS methods need to rapidly screen large number of ligands, accurately and against a large number of possible targets. Exhaustive exploration of such a vast chemical space against multiple targets is essentially impossible. Methodological innovations must be accompanied by computational infrastructure that can access libraries with representative, yet diverse chemical space ($\sim 10^{12}$ compounds).

IMPECCABLE addresses these requirements and the limitations of traditional virtual HTS [25]. It uses AI/ML methods to improve the effective sampling of individual stages, to *glue* information across different stages (e.g.. docking and MD simulations), and integrates AI/ML models with physics-based models into a single unified pipeline. IMPECCABLE uses surrogate models, which are typically computationally less expensive than the original computation, though also less accurate. Surrogate models are trained to identify or generate promising candidates, and thus IMPECCABLE is not constrained to filter a fixed set of candidates between successive stages. The effective space of ligands sampled, and thus the effective throughput is different from the number of ligands actually screened by the pipeline. In addition, information generated from one stage is used by downstream stages (e.g., ML-driven MD sampling to enhance binding free energy calculations).

Integrated AI/ML campaigns composed of workflows, which in turn are comprised of AI/ML and traditional HPC simulations, require sophisticated and scalable computational infrastructure, for which there are no turnkey or shrink-wrap solutions. IMPECCABLE employs RADICAL-Cybertools (RCT), a set of software systems, to manage the execution of heterogeneous workflows and workloads on leadership computing facilities [27]. We have previously described the infrastructure used for individual workflows [17]; here, the focus is on the computational infrastructure developed to integrate methods with varying computational characteristics into a cohesive whole to support a sustained computational campaign.

The computational campaign is part of a process—along with synthesis, experiments (e.g., biochemical or whole-cell assays), and clinic trials—whose goal is to identify promising compounds that function as COVID-19 anti-viral drugs. The campaign goal is represented by a multi-dimensional objective function which is a mix of computational and scientific objectives: optimize the number of ligands sampled while ensuring that the quality of the selected ligands is high, so as to maximize the possibility of success as an anti-viral drug. Although IMPECCABLE has identified promising leads targeted at SARS-CoV-2 [15], quantifying the scientific impact of proposed leads—global assessment or potential relative to other leads—is non-trivial, and beyond the scope of this study. We do however, highlight local enhancements—methodological and scientific. The computational objective, when articulated independent of the scientific impact, is to maximize the number of ligands screened, as well as the effective number of ligands investigated. Thus, the performance of select stages in IMPECCABLE pipeline is measured by both throughput, defined as number of ligands screened per unit

time, as well as effective throughput, defined as number of ligands sampled per unit time.

In Sec. 2 we discuss the components of the IMPECCABLE approach and how their integration delivers more than the sum of individual components. Sec. 3 outlines the diverse performance measures of the individual components, and then presents scientific results and computational performance results. Finally, we discuss the impact of IMPECCABLE and its significance beyond the specific challenge of COVID-19.

2 THE IMPECCABLE APPROACH

As shown in Fig. 1, the IMPECCABLE campaign consists of an iterative loop initiated with ML predictions (ML1), followed by three stages of data processing (S1, S2, S3). IMPECCABLE centers on the use of AI/ML techniques (ML1 and S2) interfaced with physics-based computational methods to estimate docking poses of compounds that are promising leads for a given protein target (S1) and binding free-energy computations (S3).

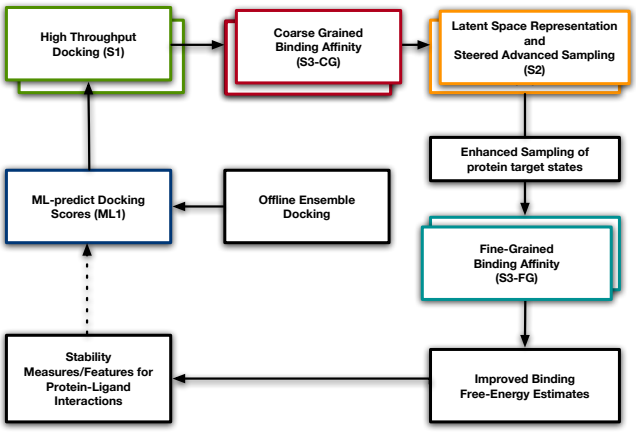

Figure 1: The IMPECCABLE Solution: represents an entire virtual drug discovery pipeline, from hit to lead through to lead optimization. The constituent components are deep-learning based surrogate model for docking (ML1), Autodock-GPU [19] (S1), coarse and fine-grained binding free energies (S3-CG and S3-FG) and S2 (DeepDriveMD).

2.1 Algorithmic & Methodological

ML1: Machine Learning Models for docking score prediction. Scoring functions are used to score poses in order to determine the most likely pose of the molecule, the magnitude of which is used to provide an indication of active versus inactive ligands, and lastly to rank order sets of libraries. We create a ML surrogate model [13] to replace the use of docking scores as a means of locating regions of chemical space likely to include strong binding drug leads. The only free variable for the surrogate ML ranking function is the basic molecular information, which typically presents as a simplified molecular-input line-entry system (SMILES) string; there is an entire field of deep learning for molecular property prediction based on this approach [16].

We use a simple featurization method, namely 2D image depictions, as this does not require complicated architectures such as graph convolution networks, while demonstrating good predictive power. From the 2D depiction of a molecule, chemists can generally identify major properties such as H-acceptors, estimate the molecule's weight, and even determine if a molecule might bind to a protein. This featurization method, unlike graph structure, is able to use off-the-shelf convolutional neural networks. By using 2D images, we are able to initialize our models with pre-trained weights that are typically scale and rotation invariant for image classification tasks, which we require in order to infer if a small molecule will bind well to a given SARS-CoV-2 target. We obtain these image depictions from the nCoV-Group Data Repository [4], which contains various descriptors generated via high-performance computations using ParSL [5] for 4.2B molecules.

We model the enrichment of our surrogate model with a regression enrichment surface (RES) [14]. The RES measures enrichment of the surrogate model, i.e., how well a surrogate model can filter molecules, measured by how many successful downstream detections are not missed by the surrogate model. The RES also models how that enrichment varies based on the threshold for filtering hits of the surrogate model. While this analysis brings forth the failure of the model to exactly replicate the rank ordering of the compounds at scale, it provides the operational benefit of these models—the predictive ML model will indeed be able to filter with near 100% accuracy two orders of magnitude more ligands from the data library. Thus, if all else is equal, with only the additional cost of docking additional compounds, we are able to expand the set of viable leads detected by two orders of magnitude, without loss of performance in the top regions of detection.

S1: High-throughput Docking. Protein-ligand docking encompasses a multistage computation consisting of ligand 3D structure (conformer) enumeration, exhaustive docking and scoring, and final pose scoring. The input to the docking protocol requires a protein structure with a designed binding region, or a crystallized ligand from which a region can be inferred, as well as a database of molecules to dock in SMILES format.

The CUDA-accelerated AutoDock 4.2.6 (AutoDockGPU) leverages a highly parallel implementation of the Lamarckian genetic algorithm (LGA) to process ligand-receptor poses in parallel over multiple compute units. AutodockGPU [19], developed in collaboration with NVIDIA and others with a target of the Summit system at the Oak Ridge Leadership Computing Facility (OLCF), applies the legacy Solis-Wets local search method along with a new local-search method based on gradients of the scoring function. One of these methods, ADADELTA [34], has proven to increase docking quality significantly in terms of RMSDs and scores, with observed speedups of 56× over the serial AutoDock 4.2 (Solis-Wets) on CPU. A drug screen takes the best scoring pose from these independent outputs. Autodock-GPU uses an OpenMP threading-based pipeline to hide ligand input and staging, and the receptor-reuse functionality for docking many ligands to a single receptor. From the computational performance perspective, we use the number of docking calculations performed per GPU as a measure of docking capability.

S2: Machine Learning Driven Molecular Dynamics. ML tools are able to quantify statistical insights concerning the time-dependent structural changes that a biomolecule undergoes in simulations, identify events that characterize large-scale conformational changes at multiple timescales, and build low-dimensional representations of simulation data capturing biophysical / biochemical / biological information. These low-dimensional representations can be used to infer kinetically and energetically coherent conformational substates and to obtain quantitative comparisons with experiments. Deep structured learning approaches automatically learn lower-level representations (or features) from input data, successively aggregating them such that they can be used in various supervised, semi-supervised, and unsupervised ML tasks.

We developed variational autoencoders to automatically reduce the high dimensionality of MD trajectories and cluster conformations into a small number of conformational states that share similar structural, and energetic characteristics [7].

We use S2 to drive adaptive sampling simulations, and use the acceleration of rare events to investigate protein-ligand interactions [18]. DeepDriveMD [9, 18] was used in S2 to simulate large ensembles of protein-ligand complexes (PLCs). DeepDriveMD builds an adaptive sampling framework to support the exploration of protein-ligand bound states that are not often accessible to approaches such as ESMACS (S3).

A key innovation is support for extremely large numbers of PLCs. This stems from the fact that a ESMACS-CG (S3-CG) simulation may generate on average, six ensembles which are analyzed by MD-driven AI approaches to identify 5–10 novel states. Hence, we also implemented a novel approach for analyzing large MD ensemble simulation datasets using a 3D adversarial autoencoder (3dAAE) [12, 33], a significant improvement over approaches such as variational autoencoders in that it is more robust and generalizable to protein coordinate datasets than contact maps (or other raw inputs) extracted from MD simulations. Similarly to autoencoders, the 3dAAE builds a latent embedding space for MD simulations to characterize conformational changes within PLCs from ESMACS-CG/FG simulation trajectories. The 3dAAE includes the PointNet encoder, Chamfer distance-based reconstruction loss, and a Wasserstein adversarial loss with gradient penalty to build a latent manifold on which all simulations are projected. From this latent manifold, we use Local Outlier Factor (LOF) [10] detection to identify 'interesting' PLCs that are then selected for S3-FG simulations. The iterative feedback between the two stages of S3-CG/FG and S2 enables accurate estimates for the binding free-energy, and allows us to filter compounds based on their affinity to the protein, while accounting for the intrinsic conformational flexibility of the PLC.

S3: Binding Free Energy Calculations. Hit-to-Lead (H2L), sometimes also called lead generation, is a step in the drug discovery process where promising lead compounds are identified from initial hits generated at preceding stages. It involves evaluation of initial hits followed by some optimization of potentially good compounds to achieve nanomolar affinities. The change in free energy between free and bound states of protein and ligand, also known as binding affinity, is a promising measure of the binding potency of a molecule, and hence it is used as a parameter for evaluating and optimizing hits at H2L stage. We employ the ESMACS (enhanced sampling of molecular dynamics with approximation of continuum solvent) protocol [30], for estimating binding affinities of PLCs. We differentiate between coarse-grained (CG) and fine-grained (FG) ESMACS variants, which differ in the number of replicas (6 vs. 24), equilibration duration (1 vs. 2ns), simulation duration (4 vs. 10ns), etc. The computational cost of ESMACS-CG is about an order of magnitude less than that of ESMACS-FG.

Binding affinity is a small number (a few tens of kcal/mol) that is derived from absolute free energies which are large (a few hundreds to thousands of kcal/mol). Thus, the usual practice of performing MMPBSA calculations on conformations generated using a single MD simulation does not give reliable binding affinities. ESMACS, on the other hand, performs ensemble MD simulation, where each independent simulation is termed a *replica*. Parameters such as the size of ensemble simulation (or the number of replicas) and the length of individual replica are chosen such that our results become reliable quantities [31]. Another factor that plays a role in determination of these parameters is the level of precision desired and the cost-benefit ratio. The number of replicas performed is adjusted to a find a sweet spot between computational cost and the level of precision acceptable at a particular stage of the pipeline.

ESMACS is costlier than the standard approach of performing a single simulation of similar duration. This increased cost, however, is more than compensated by the enhanced precision of ESMACS results which makes the resultant ranking of compounds much more reliable compared to standard approaches with similar accuracy. MMPBSA-based free energies have considerable variability in results, rendering them non-reproducible. In fact, fewer iterations are required to achieve the same level of convergence in chemical space when using ensemble simulation based methods, which leads to comparable (or even reduced) computational cost overall, than on using standard single simulation approaches. This apparent increased cost has advantages, such as increased confidence in predicted ranking of compounds, and thus more reliable training data for an ML model.

We used ESMACS-CG to perform the initial screening of thousands of hits in order to reduce computational cost while compromising on the level of precision and ranking of compounds, and used ESMACS-FG for the latter stages when we have better binding poses, and/or PLC conformations. Selectively using ESMACS-FG on a refined set of complexes decreases the computational cost substantially without affecting the quality of results.

Table 1: Normalized computational costs on Summit.

Method	Nodes per ligand	Hours per ligand (approx)	Node-hours per ligand
Docking (S1)	1/6	0.0001	~0.0001
BFE-CG (S3-CG)	1	0.5	0.5
Ad. Sampling (S2)	2	2	4
BFE-FG (S3-FG)	4	1.25	5
BFE-TI	64	10	640

2.2 Integrated Modeling Pipeline

IMPECCABLE integrates multiple virtual screening methods to select active ligands for progressively more accurate, but also more expensive, modeling. Specifically, it integrates AI/ML methods, docking methods, molecular mechanics Poisson-Boltzmann/generalized Born surface approximation (MMPBSA)-based enhanced sampling of molecular dynamics with approximation of continuum solvent (ESMACS) at the hit-to-lead stage, and Thermodynamic Integration (TI)-based Thermodynamic Integration with Enhanced Sampling (TIES) at the lead optimization stage.

At any stage, only the most promising candidates are advanced to the next stage, yielding a N-deep pipeline, where each downstream stage is computationally more expensive, but also more accurate than previous stages. The methods chosen vary in computational cost per ligand by about six orders of magnitude; in the docking stage of IMPECCABLE each dock costs about 10^{-4} node-hours per ligand; fine-grained binding free energy costs about 10^2 node-hours per ligand (Tab. 1). Tuning the cost of each method by extending or contracting the number of iterations of each method allows for enhanced scientific performance and throughput. Put together, these provide an important dynamic range of accuracy, and thus potential for scientific performance enhancement.

Each stage of IMPECCABLE when augmented with relatively simple AI/ML approaches provides a significant boost to the coverage of the compound diversity as well as conformational landscapes of PLCs. Using training data generated on small [O(10^6)] compound libraries, ML1 enables a significant improvement in filtering larger [O(10^9)] libraries, increasing coverage by 3 orders of magnitude.

The second step, which uses AutoDock-GPU results to filter the top 1% of these compounds (a ratio that can be varied by the end-user), identifies high confidence lead molecules that can bind to a given SARS-CoV-2 target. The purpose of ML1 is to predict if the given molecule will dock the protein well, and not to predict the docking pose. We exploit the intrinsic strengths of most docking programs in predicting the binding pose for a given PLC, such that the initial poses selected follow physical principles (i.e., optimizing electrostatic and hydrophobic complementarity).

The next stage, S3-CG, refines the filtered compounds to obtain an estimate of the binding-free energy. This step is crucial in the sense that it seeds the further pipeline with higher confidence leads that may have favorable interactions with the protein target.

This set of diverse PLCs are input to S2, which leverages the 3dAAE to learn a latent manifold that consists of a description of which PLCs are most stable. In addition, the latent manifold also captures intrinsic dimensions of the protein's conformational landscape that are perturbed by the ligand's interactions. Using outlier detection methods, we then filter further to include a smaller number of PLCs on which SG-FG are implemented to ultimately suggest strong confidence intervals for binding free-energy of the PLCs selected. The final stage of the pipeline provides additional features that identify key complementary features (e.g., electrostatic interactions through hydrogen bonds, or hydrophobic interactions).

IMPECCABLE embodies innovation within the individual methods it employs, as well as in the way it integrates these methods. ML techniques overcome the limitations of S1 and S3 by predicting the likelihood of binding between small molecules and a protein target (ML1), and accelerating the sampling of conformational landscapes to bound the binding free-energy values for a given PLC (S2).

Artificial intelligence (AI) and machine learning (ML) have played a pivotal role in COVID-19 drug discovery [35]. However, most AI/ML efforts have largely focused on building effective means to analyze large volumes of data generated through either ligand docking simulations—to filter favorable vs. unfavorable ligand binding poses in a given protein—or molecular dynamics (MD) simulations of selected PLCs. While docking programs are generally good at pose prediction, they are less effective in predicting binding free-energy of PLCs. Conversely, while MD simulations are effective at predicting binding-free energies, their intrinsic limitations in sampling PLC complex formation processes imply that it may be infeasible to employ them on large compound libraries.

Interfacing ML approaches with physics-based models (docking and MD simulations) has the potential to achieve at least several orders of magnitude improvement in the size of compound libraries that can be screened with traditional approaches, while simultaneously providing access to binding free-energy calculations that can impose better confidence intervals in the ligands selected for further (experimental or computational) optimization.

2.3 Computational Infrastructure

The drug candidate discovery required significant infrastructural development, performance and scale enhancement, execution optimization and unprecedented integration of diverse computational workloads/stages. The campaign employs parallelism at multiple levels to deliver these capabilities; it executes two distinct workflows sharing the same node but different workloads and task types to improve overall throughput. Further, novelty arises from a combination of scale, heterogeneous workloads, and integration of methods to work in production on leadership platforms. The campaign workload is a diverse mix of task types, e.g., MPI, single GPU, multinode GPU, and regular CPU; this mix of tasks changes over the course of the campaign. There are multiple stages that couple and concurrently execute deep learning and traditional simulations. Coupling and concurrently executing these diverse tasks is challenging, and is made more difficult by virtue of having different models and coupling with simulations across multiple stages.

The dynamic variation of workload arises due to many reasons, for example: (i) each PLC has a different rate of convergence for structural and energetic properties, and thus the duration varies; (ii) cost of docking per ligand varies across different drug compound libraries and the ligands they contain; and (iii) for methods that involve learning, (re-) training times are dependent on specific ligands and the number of simulations. The integration of diverse methods with varying computational characteristics, performance and scalability requirements, into an adaptive computational campaign requires innovative computational infrastructure.

IMPECCABLE employs the Ensemble Toolkit (EnTK) [6], which itself uses RADICAL-Pilot (RP) [21] for flexible and scalable execution of workflows with heterogeneous tasks. Together they conform to the middleware building blocks architectural pattern [27] to decouple the programming system from underlying execution capabilities. Ref. [17] provides software and infrastructure details.

2.3.1 Programming System. EnTK is a Python implementation of a workflow engine, designed to support the programming and execution of applications with ensembles of tasks. EnTK executes tasks concurrently or sequentially, depending on their arbitrary priority relation. We use the term "task" to indicate a stand-alone process that has well-defined input, output, termination criteria, and dedicated resources. For example, a task can indicate an executable which performs a simulation or a data processing analysis, executing on one or more nodes on Summit. Tasks are grouped into stages and stages into pipelines depending on the priority relation among tasks. Tasks without a reciprocal priority relation can be grouped into the same stage, whereas tasks that need to be executed before other tasks have to be grouped into different stages. Stages are then grouped into pipelines and, in turn, multiple pipelines can be executed either concurrently or sequentially. Specifically, EnTK:

- permits asynchronous execution of concurrent pipelines (each pipeline can progress at its own pace);
- allows arbitrary sizing of stages (variable concurrency);
- supports heterogeneous tasks of arbitrary types, and combinations, as well as their inter-mixing;
- promotes "ensembles" as first-class code abstraction;
- selects parameters at runtime so as to provide near-optimal selection of cost versus accuracy.

These are necessary capabilities to explore PLCs of varying complexity and cost, without constraining the number of concurrent investigations, and different methods run in arbitrary order.

2.3.2 Execution Framework for Dynamic Resource Management. Given the extreme workload heterogeneity and workload variation between and across stages, dynamic resource management is critical. Dynamic resource management capability is provided by RADICAL-Pilot (RP), a Python implementation of the pilot paradigm and architectural pattern [29]. 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 workloads, i.e., set of tasks. Tasks are executed concurrently and sequentially, depending on the available resources. For example, given 10,000 single-node tasks and 1000 nodes, a pilot system will execute 1000 tasks concurrently and each one on the remaining 9000 tasks sequentially, whenever a node becomes available. RP enables the execution of heterogeneous workloads comprised of one or more scalar, MPI, OpenMP, multi-process, and multi-threaded tasks. RP directly schedules and executes on the resources of one or more pilots without having to use the infrastructure's batch system.

RP offers unique features when compared to other pilot systems or tools that enable the execution of multi-task workloads on HPC systems: (1) concurrent execution of heterogeneous tasks on the same pilot; (2) support of all the major HPC batch systems; (3) support of more than twelve methods to launch tasks; and (4) a general purpose architecture. RP can execute single or multi core tasks within a single compute node, or across multiple nodes. RP isolates the execution of each task into a dedicated process, enabling concurrent execution of heterogeneous tasks by design.

Fig. 2 provides an overview of how the IMPECCABLE campaign is constructed and executed. It comprises four distinct computational workflows: a ML surrogate (ML1), docking (S1), binding free energy calculations (S3), and latent space representation and steered advanced sampling via MD simulations (S2). Each is a distinct workflow with well-defined inputs and outputs, multiple executables with defined dependencies, and termination criteria, able to produce stand-alone scientifically meaningful end-results. Each workflow represents the expertise and unique scientific and methodological contribution from a different team.

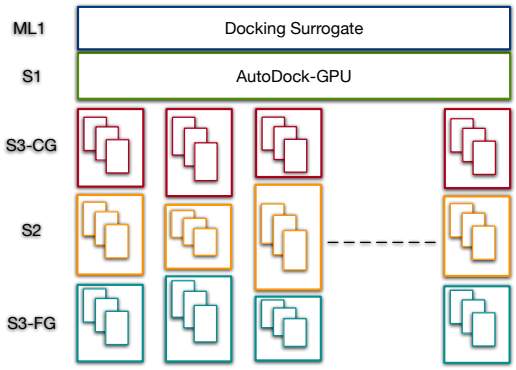

Figure 2: Programming and execution view: Each stage of the (S3-CG)-(S2)-(S3-FG) pipeline comprises multiple heterogeneous tasks; each stage executes for varying durations.

We codify IMPECCABLE workflows as a five-stage EnTK pipeline using a general-purpose language (Python) and application-specific constructs from the PST (Pipeline, Stage, Task) programming model. These abstractions simplify creating and executing ensemble applications with complex coordination and communication requirements. Pipelines can contain different workloads, e.g., distinct instances of S* for a given PLC, but also possibly multiple instances of a given S* for many PLC concurrently. Autodock-GPU is executed as a single task running on several thousand nodes, as is the docking surrogate, a relatively short duration task. The remaining three stages are workflows which are expressed as pipelines, comprised of differing stages and varying duration and number of tasks concurrently executing. The horizontal length of a box is proportional to the number of nodes used by a stage / computation, and the vertical length of boxes represent the temporal duration; boxes are not drawn to scale.

IMPECCABLE infrastructure is portable along at least three different dimensions: (i) portable to different HPC platforms, e.g., different stages of this pipeline have been run on 4 distinct leadership platforms on two continents; (ii) portable to different computations in a stage, e.g., different ML models to drive ensemble MD simulations, or different algorithms/protocols to compute free energies of binding; (iii) portable to different campaigns beyond drug selection, e.g., new materials design, catalyst discovery, or any campaign that requires the coupling of multiple levels-of-theory and accuracy.

3 PERFORMANCE AND RESULTS

Having discussed the IMPECCABLE computational campaign, we describe the composite workflows and their constituent workloads, desired performance, and factors determining scalability.

3.1 Computational Characteristics

3.1.1 ML1: Deep Learning Docking Emulator. This step is a docking emulator which serves as a pre-selection tool for docking calculations performed in step S1. The goal is to reduce the search space from about 126M ligands down to a manageable amount for the docking calculations. The emulator is based on a resnet-50 deep neural network: it transforms image representations of ligand molecules into a docking score. To convert the ligand SMILES strings into images we employed the mol2D drawing submodule from rdKit.The target scores are binding energies which are mapped into the interval [0, 1], with higher scores representing lower binding energies and thus higher docking probabilities. The main computational motifs are dense linear algebra, convolutions and elementwise operations on 4D tensors. The network is implemented in PyTorch and pre-trained on 500,000 randomly selected samples from the OZD ligand dataset across each receptor (for our purposes, each PDB entry corresponds to a separate receptor, providing access to an ensemble of docking simulations). For deployment, we compiled the model using NVIDIA TensorRT v7.2 with cuDNN v8.0 employing the `torch2trt` helper tool. As base precision we chose half precision (FP16), so that we can use the Tensor Cores on Summit's V100 GPUs.

Inference workloads are notoriously I/O bound, and thus we employ various optimizations to improve throughput. We start with the ULT911 dataset [3], which is supplied as a collection of 12,648 files with 10,000 ligands, each in Python pickle format. We first used gzip to compress each file, achieving an average compression factor of 14.2. We use MPI to distribute the individual files evenly across a large number of GPUs and bind one rank to each GPU. While we perform the model scaffolding phase, i.e. creating the computational graph and loading the weights from the pre-trained model file, each rank stages its assigned shard of the data from GPFS into node-local NVMe. During the inference process, each rank utilizes multiple data loader processes where each is employing 2 prefetching threads: the first one loads compressed files from NVMe into DRAM and decompresses them on the fly while the second iterates through the uncompressed data in memory, extracts the image and metadata information and feeds them to the neural network. The whole logic is implemented using the thread-safe Python queue module. We further use careful exception handling to make the setup resilient against sporadic I/O errors. After inference is done, the resulting lists of docking scores and metadata information such as ligand id and SMILES string are gathered and concatenated on rank 0 and written into a CSV file which is forwarded to step S1.

3.1.2 S1: Physics-based Ensemble Docking. To support the scaling requirements of S1, we implemented a Master/Worker overlay on top of the pilot-job abstraction. Fig. 3 illustrates the RAdical-Pilot Task OveRlay (RAPTOR) master/worker system as deployed on Summit. Once RAPTOR has acquired its resources by submitting a job to Summit's batch system, it bootstraps its Agent (Fig. 3-1) and then launches a task scheduler and a task executor (Fig. 3-2). Scheduler and Executor launch one or more masters on one or more compute nodes (Fig. 3-3). Once running, a master schedules one or more workers on RP Scheduler (Fig. 3-4). Those workers are then launched on one or more compute nodes by RP Executor (Fig. 3-5). Finally, the master schedules function calls on the available workers

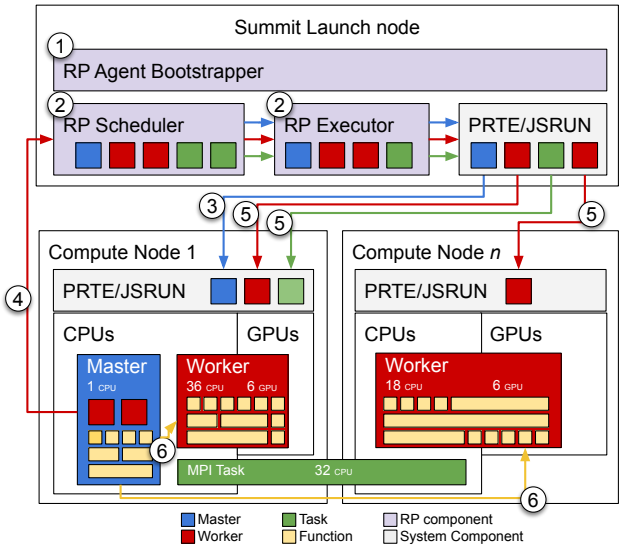

Figure 3: RAPTOR Execution Framework: One of the two execution frameworks used to support heterogeneous tasks and dynamic workloads on Summit.

for execution (Fig. 3-6), load-balancing across workers so to obtain maximal resource utilization.

The duration of the docking computation varies significantly between individual receptors. The long tail poses a challenge to load balancing; the relatively short docking times pose a challenge to scalability. Load balancing is addressed by iterating through the list of compounds in a round-robin fashion, and by dynamic load distribution which depends on the load of the individual workers. Further, balancing is achieved by: (i) tasks are communicated in bulks as to limit the communication load and frequency; (ii) multiple master processes are used to limit the number of workers served by each master, avoiding respective bottlenecks; (iii) multiple concurrent pilots are used to isolate the docking computation of individual compounds within each pilot allocation. The combination of these approaches results in a near linear scaling up to several thousand nodes, while maintaining high utilization for large numbers of concurrently used nodes. Further details can be found in Ref. [17, 21].

3.1.3 S2 and S3: Advanced Sampling and Binding Free Energy. We implement S2 and S3 as iterative pipelines that comprise heterogeneous stages, with each stage supporting the parallel execution of tasks. In S2, the pipeline starts with MD simulations that are run concurrently; it completes a single iteration by passing through deep learning stages for 3dAAE model training and the outlier detection. In a single iteration, tasks are scheduled across single GPU, multiple GPUs, and CPU-GPU tasks. For instance, the MD stage uses a single GPU per simulation (OpenMM), the data aggregation stage uses CPUs only, the ML training stage uses six GPUs per model, and the outlier detection stage uses a mixture of CPUs and GPUs. We also employ data parallelism for model training using PyTorch Distributed Data Parallel module.

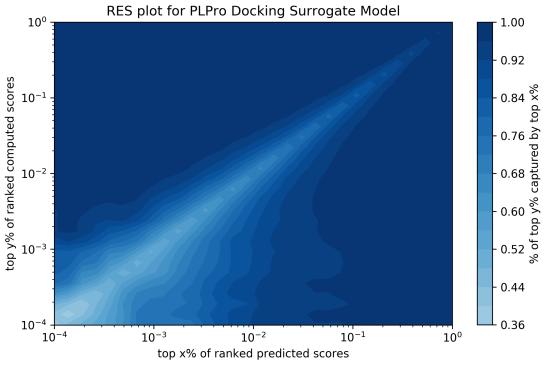

Figure 4: Regression enrichment surface (RES) profile for PLPro docking runs. As explained in the main text, RES provides a summary estimate of how many top scoring compounds can be covered given some target number (δ) of molecules to be ranked.

Similarly, S3 involves two stages of equilibration and one stage of simulation; each stage runs an ensemble of from six (S3-CG) to 24 (S3-FG) OpenMM tasks. We also employ NAMD-based TIES in conjunction with ESMACS, placing distinct simulations on GPU (OpenMM) and CPU (NAMD) concurrently to achieve optimal resource utilization on Summit.

The architecture of RAPTOR (Fig. 3) differs from that of the classic RADICAL-Pilot used for S2 and S3 on Summit [28]. The need for two task execution frameworks arises primarily from the temporal range and heterogeneity of workloads.

3.2 Scientific Results

The ZINC compound library provides over 230 million purchasable compounds [26] in ready-to-dock, 3D formats; MCULE has 100 million purchasable compounds in similar formats [3]. Hence there is a need to obtain a diverse sample of compounds for both docking calculations (to generate the training data) and inference runs (ML1 results). We selected a subset of 6.5 million "in-stock" compounds from the ZINC library along with the Enamine diversity set [1] and DrugBank compounds to develop our training library (OZD library, hereafter). We also chose a similar subset of 6.5 million compounds from the MCULE library (ORD library, hereafter) for the purposes of testing if ML1 can be used for *transferring* knowledge learned from one library to another. These libraries pay attention only to the diversity of the compounds selected, and are independently selected, although between the two libraries we observed an overlap of approximately 1.5 million compounds.

3.2.1 ML1 results. We trained our ML1 models on docking runs (generated offline) for the four main target SARS-CoV-2 proteins, namely 3C like protease (3CLPro), papain-like protease (PLPro), ADP-Ribose-1"-Monophosphatase (ADRP), and non-structural protein 15 (NSP15). These proteins all represent important drug targets against SARS-CoV-2 virus. Here we present only a vignette of results from the PLPro target and specifically from the receptor derived from the Protein DataBank IDentification (PDB ID) 6W9C. The Regression enrichment surface (RES) plot from Fig. 4 indicates the enrichment of the model as a filter where the x-axis is the cutoff for passing a molecule through the ML-filter, and the y-axis is the overall detection desired [14]. By overall detection we mean if one wishes to successfully obtain 100 compounds from a 100,000 compound library, then one wishes to identify the top 10^{-3} compounds (the overall detection desired). Given a specific budget of δ molecules to pass along—that is compounds which pass the ML-filter—we can imagine a vertical line along the x-axis of Fig. 4 at the point δ representing the budget, where any point to the right of that line represents an unattainable number of compounds. One can also imagine a constraint through $y = x$, as points above this line represent situations where a wider range of the top distribution may prove too expensive, although reasonable for some tasks. HTS is in pursuit of ultra high ranking compounds.

Given these two constraints, one can see that as δ increases, so to does the accuracy of capturing some desired threshold of the top distribution. If the library size is u, downstream tasks allot $\delta = u10^{-3}$ compounds, then the plot indicates that we will capture 50% of the top ranking $u10^{-4}$ compounds, or around 40% of the top ranking $u10^{-3}$ compounds. In concrete terms, for this library, the ML model here correctly identifies 500 of the top 1000 scoring compounds from the docking study, or about 4000 of the top 10,000 compounds. However, not all top-ranking compounds are correlated with obtaining high binding affinity to PLPro. The RES plot also provides a quantitative estimate of the number of compounds that we have to sample from lower ranking ones so that we do not inadvertently miss other high affinity compounds. Hence we also select about 15–20% of compounds from the RES for subsequent stages.

3.2.2 S3: ESMACS-CG. For each target of the four chosen proteins mentioned above, multiple crystal structures were used to perform docking, and a separate list of the top 10,000 compounds based on docking scores was generated at the ML1 stage. Therefore, depending on the number of crystal structures used for each target, there were collectively 20,000–40,000 compounds available for performing binding affinity predictions using coarse-grained ESMACS (ESMACS-CG). For this stage, we chose 10,000 compounds for each target by picking the structurally most diverse compounds. This was done for two reasons: (i) based on the docking scores, all available compounds were stable poses, and (ii) allowing for maximum possible coverage of the chemical space allowing for better and quicker identification of its relevant regions.

We performed ESMACS-CG to obtain binding affinities for all chosen compounds: a total of (four proteins) × (10,000 compounds) = 40,000 S3-CG calculations. These values (e.g., see Fig. 5A for a probability distribution of the 10,000 binding affinities computed for PLPro) typically lie between -60 to +20 kcal/mol. The resultant trajectories and binding affinity values from this stage were used as input for S2 to identify potentially useful conformations that were fed into S3-FG.

3.2.3 Using S2 to seed S3-FG. Measuring DeepDriveMD performance for PLCs presents challenges. The input from the S3-CG

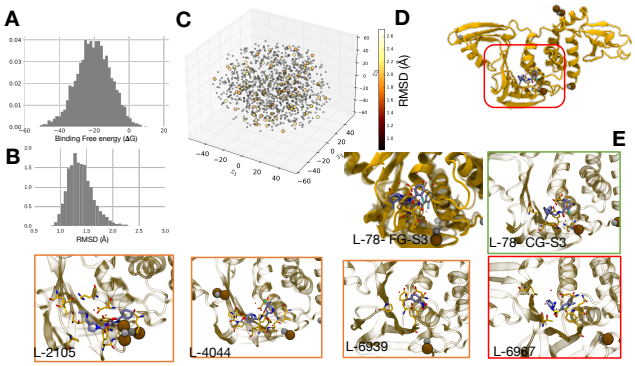

Figure 5: Initial results from IMPECCABLE on PLPro receptor (PDB ID: 6W9C). (A) Summary histogram of the distribution of binding free energies estimated using ESMACS-CG. (B) Summary of RMSD (Å) determined from ESMACS-CG PLC ensembles show a rather tight distribution with a few PLCs that exhibit greater fluctuations. (C) Latent space representation from the 3dAAE model depicting the outliers from RMSD distributions (>1.9 Å) and the rest as gray dots. The latent space also summarizes the extent of sampling from these simulations. (D) Structure of PLPro bound to one of the highly specific molecule (L78) in its active site. (E) A zoomed in version of the same compound (L78) showing close interactions with key residues in PLPro (green highlight). The panel on the left depicts how upon running ESMACS-FG we obtain tighter binding through the compound moving further into the binding site, forming strong hydrophobic interactions and hydrogen bonds.

pipeline stage are relatively short time-scale, whereas PLC association processes tend to vary significantly in time scales. Thus, we chose a pragmatic measure of PLC stability that takes into account the number of heavy atom contacts between the protein and the ligand of interest. From the top scoring PLCs that are selected from S3-CG, we use the 3dAAE and LOF to filter those conformations that show increased stability profiles in the PLCs. We believe these PLCs are of the most interest, since the increased stability potentially contributes to favorable interactions between protein and ligand. We also measure the 3dAAEs performance in terms of its ability to learn effective latent space representations from S3-CG stage (through standard measures such as training and validation loss metrics).

For PLPro, about 5000 compounds were chosen based on the structural diversity criterion for Protein Data Bank (PDB) ID 6W9C. The trajectories corresponding to these 5000 ligands generated by S3-CG were used to build a combined dataset of 100,978 examples. The point cloud data, representing the coordinates of the 309 backbone C^{α} atoms of the protein, was randomly split into training (80%) and validation input (20%) and was used to train the 3dAAE model for 100 epochs using a batch size of 64. The data was projected onto a latent space of 64 dimensions constrained by a Gaussian prior distribution with a standard deviation of 0.2. The loss optimization was performed with the Root Mean Square Propagation (RMSprop) optimizer, a gradient descent algorithm for mini-batch learning, using a learning rate of 0.00001.

We also added hyperparameters to scale individual components of the loss. The reconstruction loss was scaled by 0.5 and the gradient penalty was scaled by a factor of 10. We trained the model using several combinations of hyperparameters, mainly varying learning rate, batch size and latent dimension. The embedding learned from the 3dAAE model summarizes a latent space that is similar to variational autoencoders, except that 3dAAEs tend to be more robust to outliers within the simulation data. The embeddings learned from the simulations allow us to cluster the conformations (in an unsupervised manner) based on their similarity in overall structure, which can be typically measured using quantities such as root-mean squared deviations (RMSD). The 5,000 ligands were further analyzed and 5 structures with the lowest free energy (L6967, L2105, L78, L6939, L4044) were selected for generating embeddings for 1200 examples, using the hyperparameters learned from 3dAAE performed on the full set of 5,000 ligands. For visualizing and assessing the quality of the model in terms latent space structure, we computed t-stochastic neighborhood embedding (t-SNE) [20] on the validation embeddings. The validation data was painted with grey while the test data was painted with the RMSD of each structure to the starting conformation (Fig. 5B-C).

3.2.4 S3: ESMACS-FG. The large amount of data generated by S3-CG was analyzed at S2, from which, potentially good conformations were identified for compounds with large negative binding affinities from ESMACS-FG. This process led us to filter out five outlier conformations each for the top five compounds based on S3-CG results. We used these 25 conformations to perform the costlier fine-grained ESMACS (ESMACS-FG) calculations to investigate if IMPECCABLE pipeline can identify favorable interactions between protein and ligands. If so, it would help identify favorable regions in the chemical space deserving more attention, which in turn trains our ML model to generate and/or predict better compounds in subsequent iteration.

Fig. 6 shows that ESMACS-FG predicts much lower binding affinities than those predicted by ESMACS-CG (Fig. 5D-E). The force-field used in both cases was the same; only the starting structures varied. This implies that the outliers filtered by S2 indeed captured some favorable interactions and successfully identified good conformations out of the large number of conformations generated by S3-CG. This is indicative of a novel capability of IMPECCABLE to quickly sample the relevant chemical space and hence accelerate the process of drug discovery.

3.3 Computational Performance

Fig. 7 shows an example of how independent pipelines can be integrated into a single workflow. Each pipeline is comprised of stages, each with an arbitrary number of tasks. Tasks have heterogeneous execution time and computational requirements. Stages can execute concurrently or sequentially, depending on available resources and task, stage and pipeline interdependencies. In the depicted integration, single-GPU tasks execute alongside MPI GPU and few CPU tasks, in distinct and customized execution environments. Note that the overheads (light-colored vertical areas of the plots) are invariant

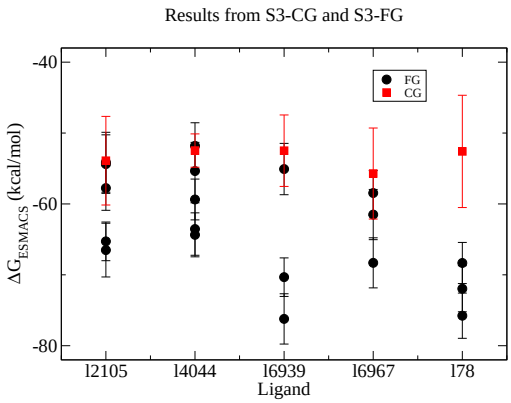

Figure 6: Comparison of S3-CG and S3-FG results for the five best binders for PLPro (PDB ID: 6W9C) based on ESMACS-CG results. S2 selected five outlier conformations for each binder and performed ESMACS-FG on them. Initial results indicate improved binding for the selected conformations in all five compounds, as FG energies are lower than CG.

to scale, i.e., they do not depend on the number of concurrent tasks executed or on the length of those tasks.

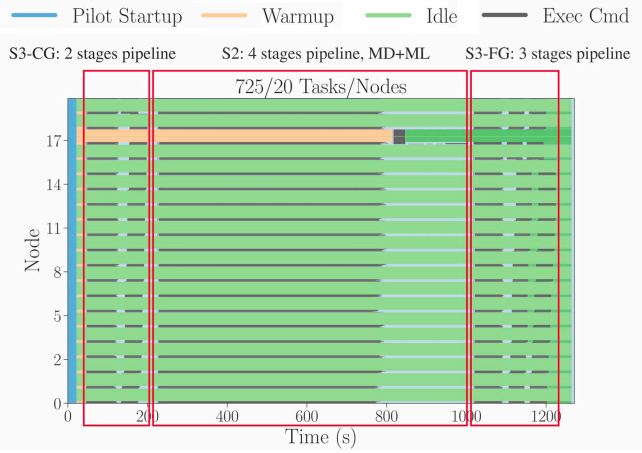

Figure 7: A time-series of node utilization for the integrated execution of three GPU-intensive workflows (S3-CG)-(S2)-(S3-FG). S3-CG, S2 and S3-FG are heterogeneous and multi-stage workflows themselves.

We measure flops (floating point operations, not rates) per work unit for the most relevant components of each stage. We define a *work unit* to be a representative code section such as an MD time integration step for MD-based or a data sample for DL-based applications. Thus we can compute the aggregate invested flops by scaling the measured flop counts to the respective work set sizes used in the actual runs.

We always normalize the measurements to a single Summit node for the same reason. As all of our applications are perfectly load balanced with respect to a Summit node (mostly even with respect to individual GPUs within a node), this procedure yields a representative flop count. We use the methodology of Yang et al. [32] and the NVIDIA NSight Compute 2020 GPU profiling tool to measure flops for all precisions and sum them to obtain a mixed precision flop count. When possible, we use start/stop profiler hooks to filter out the representative work units. In order to obtain the flop rate, we divide the aggregated flops for each EnTK task by the time it takes to complete that respective task, including pre- and post-processing overhead. Note that we do not account for any CPU flops invested in this calculation as we expect that number to be small. We discuss the specifics for each component:

ML1: We count flops as described above for 10 steps at batch size 256. From that, we derive a flop count per batch per GPU.

S1: We count flops for a five-ligand AutoDock-GPU run on one GPU to derive flops for a single ligand. We chose this ligand complexity to represent the majority of the ligands processed in the run.

S2: This stage has multiple steps, but we only account for the autoencoder training and the MD performed in this stage. For the former, we measure the flops per batch for a batch size of 32 for training and validation separately and weight them proportionally by their relative number of batches. After each training epoch, a validation is performed and the train/validation dataset split is 80%/20%. This can be translated into a overall flop count for the full autoencoder stage. For the MD part, we profile 20 steps of OpenMM and compute a complexity per step.

S3-CG/FG: These two stages both have two steps, a minimization and an MD step. We count the flops for 10 iterations of the minimization algorithm and for 20 steps of the MD run to derive a flop count per minimization and MD step. Since the algorithmic complexity differs between CG and FG, we profile those separately.

As shown in Fig. 7 and Tab. 2, both the ML and ensemble simulations (S3-FG) approach 1 Pflop/s sustained [13, 21]; there are no middleware barriers to utilizing all of Summit (or any other leadership class machine).

Table 2: Peak flop per second (mixed precision, measured over short but time interval) and Peak throughput (number of ligands processed per second) for different stages, for given node (GPU) counts on Summit.

Comp.	GPUs	Tflop/s	Throughput (ligands/s)
ML1	1536	753.9	319674 ligands/s
S1	6000	112.5	14252 ligands/s
S3–CG	6000	277.9	2000 ligand/s
S3–FG	6000	732.4	200 ligand/s

4 DISCUSSION AND CONCLUSIONS

Multi-scale biophysics-based computational lead discovery is an important strategy for drug development. Until now it has been slower than experimental screening, and at insufficient scale to explore libraries of billions of molecules, even on the most powerful machines. The work reported here addresses these dual issues by integrating ML components with physics-based components, and executing them at scale on leadership-class platforms.

IMPECCABLE implements and leverages multi-level parallelism: concurrent pipelines composed of multiple stages; each stage composed of multiple tasks. Each IMPECCABLE stage is distinct, and has different computational workloads: high-throughput functions, pure ML, hybrid ML-HPC, and ensemble-MD workloads. Different stages use either all, or large fractions, of the largest leadership class machines. IMPECCABLE integrates these heterogeneous workflows into a unified campaign for computational lead discovery. The scale and integration across workload heterogeneity is important. Scale is critical for number of drug candidates screened, accuracy and confidence of results, and to generate training data fast enough to use ML-based surrogate models [13]. The integration of methods supported by scalable infrastructure, speeds-up drug-discovery by orders of magnitudes.

As an illustrative example, the work reported here has played a central role in DOE's National Virtual Biotechnology Laboratory (NVBL) [2] effort to use computational molecular design approaches to develop medical therapeutics for COVID-19 [17]. Methods and infrastructure reported in this paper have been used to screen over 4.2 billion molecules [4] against over a dozen drug targets in SARS-CoV-2, leading to the identification and experimental validation of over 1000 compounds, resulting in over 40 hits that are progressing to advanced testing [15]. This screening used more than 5.0M node-hours across diverse HPC platforms. Individual parts of the campaign have been used on ~7000 nodes, and used to sustain 144M/hour docking hits [15]. In doing so, IMPECCABLE has screened $\sim 10^{11}$ ligands. IMPECCABLE has computed binding free energies on 10^4 PLCs concurrently. Individual workflow components deliver 100× to 1000× improvement over traditional methods [9]. Methodological advances and scale of execution has enabled the NVBL to discover a promising anti-viral drug candidate [15].

While much work remains to be done, we have demonstrated some important milestones towards the ultimate goal of efficient and high-throughput virtual screening. Put together, this work demonstrates a path towards a 1000× improvement of overall throughput of computational drug discovery [9, 13, 15]. This makes the search of giga-scale libraries of compounds across collections of drug targets feasible and routine. IMPECCABLE has developed the necessary infrastructure [17] to support the integration of physics-based modeling with AI methods into a single campaign comprised of multiple workflows [15] at the largest possible scale.

IMPECCABLE represents the first-step towards framing the selection of leads as a complex design problem. The multi-stage pipeline can be formulated as an optimal design of experiments, with selection percentage, computational cost versus uncertainty of different computational stages, quality of selection, inter alia as campaign degrees-of-freedom. Diverse approaches from Bayesian optimization, multi-arm bandit, and mean objective cost of uncertainty are being used to optimally select, given a certain computational budget or other constraint, the most promising leads. These methods will not change the raw throughput, but will impact the effective throughput.

IMPECCABLE infrastructure can be adapted to a broad range of near autonomous drug development scenarios by means of additional modules and models that fill out the drug discovery pipeline. For example, this campaign will impact multiple target problems, including the DOE/NCI JDAS4C Pilot1 effort to advance Cancer drug development through AI and the related ECP CANDLE project. We are building Cancer drug response models that predict the response of tumors to drugs or drug combinations, which will be coupled to similar campaigns, to add additional feedback on predicted efficacy of a target molecule.

ACKNOWLEDGMENTS

Research was supported by the DOE Office of Science through the National Virtual Biotechnology Laboratory, a consortium of DOE national laboratories focused on response to COVID-19, with funding provided by the Coronavirus CARES Act. This research was supported as part of the CANDLE project by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration. This work has been supported in part by the Joint Design of Advanced Computing Solutions for Cancer (JDACS4C) program established by the U.S. Department of Energy (DOE) and the National Cancer Institute (NCI) of the National Institutes of Health. We are grateful for funding for the UK MRC Medical Bioinformatics project (grant no. MR/L016311/1), the UK Consortium on Mesoscale Engineering Sciences (UKCOMES grant no. EP/L00030X/1) and the European Commission for the EU H2020 CompBioMed2 Centre of Excellence (grant no. 823712), as well as financial support from the UCL Provost. Access to SuperMUC-NG, at the Leibniz Supercomputing Centre in Garching, was made possible by a special COVID-19 allocation award from the Gauss Centre for Supercomputing in Germany. Anda Trifan acknowledges support from the United States Department of Energy through the Computational Sciences Graduate Fellowship (DOE CSGF) under grant number: DE-SC0019323. We acknowledge amazing support from OLCF— Don Maxwell, Bronson Messier and Sean Wilkinson. We also wish to thank Dan Stanzione and Jon Cazes at Texas Advanced Computing Center.

REFERENCES

[1] [n.d.]. Enamine diversity library. https://enamine.net/hit-finding/diversity-libraries.
[2] [n.d.]. National Virtual Biotechnology Laboratory (NVBL). https://science.osti.gov/nvbl.
[3] [n.d.]. Ultimate 100 Million Compounds. https://ultimate.mcule.com.
[4] Yadu Babuji, Ben Blaiszik, Tom Brettin, Kyle Chard, Ryan Chard, Austin Clyde, Ian Foster, Zhi Hong, Shantenu Jha, Zhuozhao Li, et al. 2020. Targeting SARS-CoV-2 with AI-and HPC-enabled lead generation: A First Data release. *arXiv preprint arXiv:2006.02431* (2020).
[5] Yadu Babuji, Anna Woodard, Zhuozhao Li, Daniel S Katz, Ben Clifford, Rohan Kumar, Lukasz Lacinski, Ryan Chard, Justin M Wozniak, Ian Foster, and Kyle Chard. 2019. Parsl: Pervasive parallel programming in Python. In *28th International Symposium on High-Performance Parallel and Distributed Computing*. 25–36.
[6] Vivek Balasubramanian, Matteo Turilli, Weiming Hu, Matthieu Lefebvre, Wenjie Lei, Ryan Modrak, Guido Cervone, Jeroen Tromp, and Shantenu Jha. 2018. Harnessing the power of many: Extensible toolkit for scalable ensemble applications. In *International Parallel and Distributed Processing Symposium*. IEEE, 536–545.
[7] Debsindhu Bhowmik, Shang Gao, Michael T. Young, and Arvind Ramanathan. 2018. Deep clustering of protein folding simulations. *BMC Bioinformatics* 19, 18 (2018), 484. https://doi.org/10.1186/s12859-018-2507-5
[8] Regine S Bohacek, Colin McMartin, and Wayne C Guida. 1996. The art and practice of structure-based drug design: A molecular modeling perspective. *Medicinal research reviews* 16, 1 (1996), 3–50.
[9] Alexander Brace, Hyungro Lee, Heng Ma, Anda Trifan, Matteo Turilli, Igor Yakushin, Todd Munson, Ian Foster, Shantenu Jha, and Arvind Ramanathan. 2021. Achieving 100X faster simulations of complex biological phenomena by coupling ML to HPC ensembles. arXiv:2104.04797 [cs.DC]

[10] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *ACM SIGMOD International Conference on Management of Data*. 93–104.

[11] JR Broach and J Thorner. 1996. High-throughput screening for drug discovery. *Nature* 384, 6604 Suppl (Nov. 1996), 14—16. https://doi.org/10.1038/384014a0

[12] Lorenzo Casalino, Abigail Dommer, Zied Gaieb, Emilia P. Barros, Terra Sztain, Surl-Hee Ahn, Anda Trifan, Alexander Brace, Anthony Bogetti, Heng Ma, Hyungro Lee, Matteo Turilli, Syma Khalid, Lillian Chong, Carlos Simmerling, David J. Hardy, Julio D. C. Maia, James C. Phillips, Thorsten Kurth, Abraham Stern, Lei Huang, John McCalpin, Mahidhar Tatineni, Tom Gibbs, John E. Stone, Shantenu Jha, Arvind Ramanathan, and Rommie E. Amaro. 2020. AI-Driven Multiscale Simulations Illuminate Mechanisms of SARS-CoV-2 Spike Dynamics. (2020). https://doi.org/10.1101/2020.11.19.390187

[13] Austin Clyde, Thomas Brettin, Alex Partin, Hyunseung Yoo, Yadu Babuji, Ben Blaiszik, Andre Merzky, Matteo Turilli, Shantenu Jha, Arvind Ramanathan, and Rick Stevens. 2021. Protein-Ligand Docking Surrogate Models: A SARS-CoV-2 Benchmark for Deep Learning Accelerated Virtual Screening. arXiv:2106.07036 [q-bio.BM]

[14] Austin Clyde, Xiaotian Duan, and Rick Stevens. 2020. Regression Enrichment Surfaces: a Simple Analysis Technique for Virtual Drug Screening Models. *arXiv preprint arXiv:2006.01171* (2020).

[15] Austin Clyde, Stephanie Galanie, Daniel W Kneller, Heng Ma, Yadu Babuji, Ben Blaiszik, Alexander Brace, Thomas Brettin, Kyle Chard, Ryan Chard, et al. 2021. High Throughput Virtual Screening and Validation of a SARS-CoV-2 Main Protease Non-Covalent Inhibitor. *bioRxiv* (2021).

[16] Daniel C. Elton, Zois Boukouvalas, Mark D. Fuge, and Peter W. Chung. 2019. Deep learning for molecular design—a review of the state of the art. *Molecular Systems Design & Engineering* 4, 4 (2019), 828–849. https://doi.org/10.1039/c9me00039a

[17] Hyungro Lee, Andre Merzky, Li Tan, Mikhail Titov, Matteo Turilli, Dario Alfe, Agastya Bhati, Alex Brace, Austin Clyde, Peter Coveney, Heng Ma, Arvind Ramanathan, Rick Stevens, Anda Trifan, Hubertus Van Dam, Shunzhou Wan, Sean Wilkinson, and Shantenu Jha. 2021. Scalable HPC and AI Infrastructure for COVID-19 Therapeutics. *Platform for Advanced Scientific Computing (PASC)* (2021). https://doi.org/10.1145/3468267.3470573. https://arxiv.org/abs/2010.10517.

[18] Hyungro Lee, Matteo Turilli, Shantenu Jha, Debsindhu Bhowmik, Heng Ma, and Arvind Ramanathan. 2019. DeepDriveMD: Deep-Learning Driven Adaptive Molecular Simulations for Protein Folding. In *2019 IEEE/ACM Third Workshop on Deep Learning on Supercomputers (DLS)*. IEEE, 12–19. https://doi.org/10.1109/DLS49591.2019.00007 arXiv:1909.07817

[19] Scott LeGrand, Aaron Scheinberg, Andreas F Tillack, Mathialakan Thavappiragasam, Josh V Vermaas, Rupesh Agarwal, Jeff Larkin, Duncan Poole, Diogo Santos-Martins, Leonardo Solis-Vasquez, et al. 2020. GPU-Accelerated Drug Discovery with Docking on the Summit Supercomputer: Porting, Optimization, and Application to COVID-19 Research. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*. 1–10.

[20] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[21] Andre Merzky, Matteo Turilli, Mikhail Titov, Aymen Al-Saadi, and Shantenu Jha. 2021. Design and Performance Characterization of RADICAL-Pilot on Leadership-class Platforms. *Accepted for Transactions of Parallel and Distributed Computing (TPDS), Special Issue on R&D for ExaScale Computing, arXiv preprint arXiv:2103.00091* (2021).

[22] Jerry M. Parks and Jeremy C. Smith. 2020. How to Discover Antiviral Drugs Quickly. *New England Journal of Medicine* 382, 23 (2020), 2261–2264. https://doi.org/10.1056/NEJMcibr2007042 PMID: 32433861.

[23] Edward O Pyzer-Knapp, Changwon Suh, Rafael Gómez-Bombarelli, Jorge Aguilera-Iparraguirre, and Alán Aspuru-Guzik. 2015. What is high-throughput virtual screening? A perspective from organic materials discovery. *Annual Review of Materials Research* 45 (2015), 195–216.

[24] Atanu Saha and Heather Roberts. 2020. Pharmaceutical industry's changing market dynamics. *International Journal of the Economics of Business* (2020), 1–17.

[25] Gisbert Schneider. 2010. Virtual screening: an endless staircase? *Nature Reviews Drug Discovery* 9, 4 (2010), 273–276.

[26] Teague Sterling and John J. Irwin. 2015. ZINC 15 – Ligand Discovery for Everyone. *Journal of Chemical Information and Modeling* 55, 11 (2015), 2324–2337.

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

[28] Matteo Turilli, Andre Merzky, Thomas Naughton, Wael Elwasif, and Shantenu Jha. 2019. Characterizing the Performance of Executing Many-tasks on Summit. *IPDRM Workshop, SC19* (2019). https://arxiv.org/abs/1909.03057.

[29] Matteo Turilli, Mark Santcroos, and Shantenu Jha. 2018. A Comprehensive Perspective on Pilot-Job Systems. *ACM Comput. Surv.* 51, 2, Article 43 (April 2018), 32 pages. https://doi.org/10.1145/3177851

[30] Shunzhou Wan, Agastya P. Bhati, Stefan J. Zasada, Ian Wall, Darren Green, Paul Bamborough, and Peter V. Coveney. 2017. Rapid and Reliable Binding Affinity Prediction of Bromodomain Inhibitors: A Computational Study. *Journal of Chemical Theory and Computation* 13, 2 (2017), 784–795. https://doi.org/10.1021/acs.jctc.6b00794 arXiv:https://doi.org/10.1021/acs.jctc.6b00794 PMID: 28005370.

[31] David W. Wright, Fouad Husseini, Shunzhou Wan, Christophe Meyer, Herman van Vlijmen, Gary Tresadern, and Peter V. Coveney. 2020. Application of the ESMACS Binding Free Energy Protocol to a Multi-Binding Site Lactate Dehydogenase A Ligand Dataset. *Advanced Theory and Simulations* 3, 1 (2020), 1900194. https://doi.org/10.1002/adts.201900194

[32] Charlene Yang. 2020. Hierarchical Roofline Analysis: How to Collect Data using Performance Tools on Intel CPUs and NVIDIA GPUs. arXiv:2009.02449 [cs.DC]

[33] Maciej Zamorski, Maciej Zięba, Piotr Klukowski, Rafał Nowak, Karol Kurach, Wojciech Stokowiec, and Tomasz Trzciński. 2018. Adversarial Autoencoders for Compact Representations of 3D Point Clouds. *arXiv preprint arXiv:1811.07605* (2018).

[34] Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* (2012).

[35] Yadi Zhou, Fei Wang, Jian Tang, Ruth Nussinov, and Feixiong Cheng. 2020. Artificial intelligence in COVID-19 drug repurposing. *The Lancet Digital Health* (2020).