MOFA: Discovering Materials for Carbon Capture with a GenAI- and Simulation-Based Workflow

Xiaoli Yan*†#, Nathaniel Hudson*‡#, Hyun Park*§, Daniel Grzenda‡, J. Gregory Pauloski‡, Marcus Schwarting‡, Haochen Pan‡, Hassan Harb*, Samuel Foreman*, Chris Knight*, Tom Gibbs¶, Kyle Chard*‡, Santanu Chaudhuri*†, Emad Tajkhorshid§, Ian Foster*‡, Mohamad Moosavi‖, Logan Ward*, E. A. Huerta*‡§

*Argonne National Laboratory; Lemont, IL, United States
†University of Illinois Chicago; Chicago, IL, United States
‡University of Chicago; Chicago, IL, United States
§University of Illinois Urbana-Champaign; Urbana, IL, United States
¶NVIDIA Inc.; Santa Clara, CA, United States
‖University of Toronto; Toronto, Ontario

***Abstract*—We present MOFA, an open-source generative AI (GenAI) plus simulation workflow for high-throughput generation of metal-organic frameworks (MOFs) on large-scale high-performance computing (HPC) systems. MOFA addresses key challenges in integrating GPU-accelerated computing for GPU-intensive GenAI tasks, including distributed training and inference, alongside CPU- and GPU-optimized tasks for screening and filtering AI-generated MOFs using molecular dynamics, density functional theory, and Monte Carlo simulations. These heterogeneous tasks are unified within an online learning framework that optimizes the utilization of available CPU and GPU resources across HPC systems. Performance metrics from a 450-node (14,400 AMD Zen 3 CPUs + 1800 NVIDIA A100 GPUs) supercomputer run demonstrate that MOFA achieves high-throughput generation of novel MOF structures, with CO_2 adsorption capacities ranking among the top 10 in the hypothetical MOF (hMOF) dataset. Furthermore, the production of high-quality MOFs exhibits a linear relationship with the number of nodes utilized. The modular architecture of MOFA will facilitate its integration into other scientific applications that dynamically combine GenAI with large-scale simulations.**

***Index Terms*—Generative AI, High Throughput Workflow, Heterogeneous Computing, Online Learning, Atomistic Simulations, Metal-Organic Frameworks, Carbon Capture**

I. Introduction

Carbon dioxide (CO_2) has been identified as the primary contributor to the elevation of earth's atmospheric temperature [1]. The combustion of fossil fuel is the major source of CO_2 emission. CO_2 emitted and absorbed by non-human activity can be a self-contained cycle. Developing new technologies to capture CO_2 from the human activity can be one of the most efficient solutions to mitigate the global climate change.

Metal-Organic Frameworks (MOFs) are materials that have drawn much attention in the scientific community for their potential in numerous applications, including carbon capture [2], [3]. MOFs typically comprise two main types of components: *(i)* an organic molecule ("linker" or "ligand"; synonymous) and *(ii)* an inorganic metal ("cluster"), organized in a topology that can allow them, for example, to store gases like hydrogen or carbon dioxide [4]. This ability makes MOFs exciting for applications other than carbon capture such as catalysis [5], drug discovery [6], and luminescence sensing [7].

Generative AI (GenAI) methods are by now widely used to rapidly generate fluid text and high-resolution images. Image generative tools using denoising diffusion models (e.g., Stable Diffusion [8]) can generate photo-realistic images in response to a text prompt. In materials science, diffusion models can be trained instead to generate novel molecular structures with target chemical properties as prompts [9]–[11]. But there remain challenges regarding how to *(i)* intelligently traverse chemical space and *(ii)* efficiently execute large-scale, high-throughput MOF generative workflows at scale.

A simple brute-force approach to creating many new MOF structures would be to combine metal nodes with organic linkers in different geometries exhaustively. However, the chemical search space for MOFs is intractably large due to the many possible metal nodes, organic linkers, and pore geometries [12]. The use of a generic chemical GenAI model to produce linkers is also infeasible, as the resulting molecules are not more likely to produce better MOFs than those found with brute-force searches. We instead need a GenAI that is fine-tuned to generate those rare molecules that yield MOF structures with *interesting* chemical properties—thus requiring fewer guesses to reach the same answer.

We approach this task of producing a GenAI model able to efficiently explore the large space of possible MOFs as follows. First, we construct an initial GenAI model for MOF linker generation, MOFLinker, by fine-tuning an existing model developed for drug discovery on subspaces of high-performing MOFs. Then, we refine MOFLinker over time by repeating steps: *(i)* **generate** new linkers with MOFLinker, *(ii)* **assemble** new MOFs from generated linkers, *(iii)* **screen** assembled MOFs to eliminate non-promising linkers, and *(iv)* **retrain** MOFLinker using performant linkers identified through screening with the goal of improving the quality of linkers produced. The screening step, in particular, is crucial to the success of this workflow. Screening combines simple structural tests of

#Equal contribution.

MOF feasibility with the use of expensive quantum chemistry tools (CP2K [13], LAMMPS [14], RASPA [15]) to compute important MOF properties such as stability and porosity.

Our workflow thus combines GenAI, in the form of MOFLinker, with both molecule assembly and a variety of screening computations to navigate chemical space. This combination of elements makes efficient workflow execution challenging. First, massively parallel high-performance computing (HPC) is needed to enable rapid screening of many candidates molecules. Second, tasks in the workflow have heterogeneous hardware requirements, with different tasks executing most efficiently on CPUs (e.g., molecule structure assembly), GPUs (e.g., GenAI training and inference), or a mix of both (e.g., quantum chemistry simulations). This diversity poses a heterogeneous computing challenge: a high-throughput MOF generation workflow needs to create new and relevant MOF structures while coordinating the heterogeneous needs of the different task types. Ideally, such a workflow will produce more novel, valid, and relevant MOF structures when consuming more node-hours on an HPC system.

To address these challenges, we introduce MOFA, an open-source computational workflow that uses online learning to generate novel MOF structures for carbon capture.[1] We focus the design of MOFA around the objective of generating MOFs for carbon capture. At a high level, MOFA is a heterogeneous workflow consisting of two task types: *(i)* GPU-optimized tasks for inference and training of a GenAI diffusion model for generating the building blocks of new MOFs and *(ii)* CPU- and GPU-optimized tasks that screen generated MOFs based on chemical properties calculated through atomistic simulations. To dynamically schedule tasks with varying resource requirements, MOFA is built on the Parsl [16] and Colmena frameworks [17] to scale the workload across heterogeneous resources on large-scale computational systems.

The central contributions of this paper are as follows:

1) We describe an HPC-coupled-generative AI workflow, MOFA, for high-throughput discovery of MOFs for carbon capture on heterogeneous HPC systems. Its open source, modular implementation will facilitate its use for both computer science research and other applications that combine GenAI and large scale simulations.
2) We demonstrate, via a 450-node (14,400 AMD Zen 3 CPUs + 1800 NVIDIA A100 GPUs), 3-hour run, that MOFA can produce 114 MOFs per hour with competitive CO_2 adsorption capacities, with one MOF in the top 5 (4.05 mol/kg at 0.1 bar) and ten MOFs in the top 10% (1–2 mol/kg at 0.1 bar) of the 4547-MOF structurally similar subset of the 137,652-MOF hMOF dataset [18].
3) We demonstrate in this and other runs, plus supporting analyses, that MOFA achieves high computational efficiency on modern GPU- and CPU-based HPC systems.

II. Related Work

Current approaches to accelerating the rational discovery of MOFs combine one or several of the following methods: AI, high throughput screening, and atomistic simulations. Here, we discuss related work in MOF discovery and systems for executing heterogeneous computing workflows.

A. MOFs & Their Discovery

MOFs are versatile materials composed of metal ion clusters coordinated with organic linkers to form porous crystalline structures [19], [20]. Their tunable pore sizes, high surface areas, and structural flexibility have attracted significant attention since the 1990s, enabling them to serve in a range of applications such as gas storage, separation, and catalysis [21]–[24].

Prior to the adoption of GenAI approaches, MOF screening workflows often used brute-force [25], [26], heuristics-driven [3], [27], or sampling strategies [28] to define and prioritize candidate MOF structures. GenAI models, which are capable of producing novel experiments based on previous successful attempts, can augment these existing workflow strategies. Model architectures such as diffusion, generative adversarial networks, and variational autoencoders have been employed in applications such as de novo drug screening [29], shape optimization [30], chemical synthesis identification [31], and drug discovery [32].

In the context of MOF design, MOFDiff [33] is a coarse-grained diffusion architecture developed to produce MOF structures with effective CO_2 separation capabilities. MOFDiff starts with a coarse-grained representation of a MOF (comprised of nodes and connecting linkers), and diffusion-based denoising supplies a refined, full-atom representation which can be assessed with various simulation methods. Similarly, SmVAE [34] introduces a variational autoencoder that effectively encodes MOF building blocks and stochastically decodes novel MOF structures which are targeted for higher CO_2 capacity. Finally, GHP-MOFassemble is a fine-tuned version of the DiffLinker architecture (a GenAI model originally trained for drug design and discovery) meant for the production of *de novo* linkers for MOFs [11]. For our proposed MOFA, we approach GenAI-driven MOF discovery by fine-tuning the DiffLinker architecture in a high throughput, online learning workflow which periodically re-trains over time.

B. Heterogeneous Computing Workflows

The demands of scientific workflows that couple AI methods with simulation tools [35] has spurred the use of heterogeneous hardware within a single application. Simulation code may need many CPU and/or GPU cores across many nodes, while AI models are executed most efficiently on specialized accelerators (e.g., GPUs, wafer-scale systems [36]) with high-bandwidth memory. Further challenges include flexible routing of results between different components, re-allocation of resources between different task types, and reducing workflow latencies so systems can respond quickly to new information.

Workflow systems enable the expression and execution of applications composed of multiple distinct task types. The

[1] Repository for MOFA can be found here: https://github.com/XXX/YYY (redacted to adhere to double-blind review).

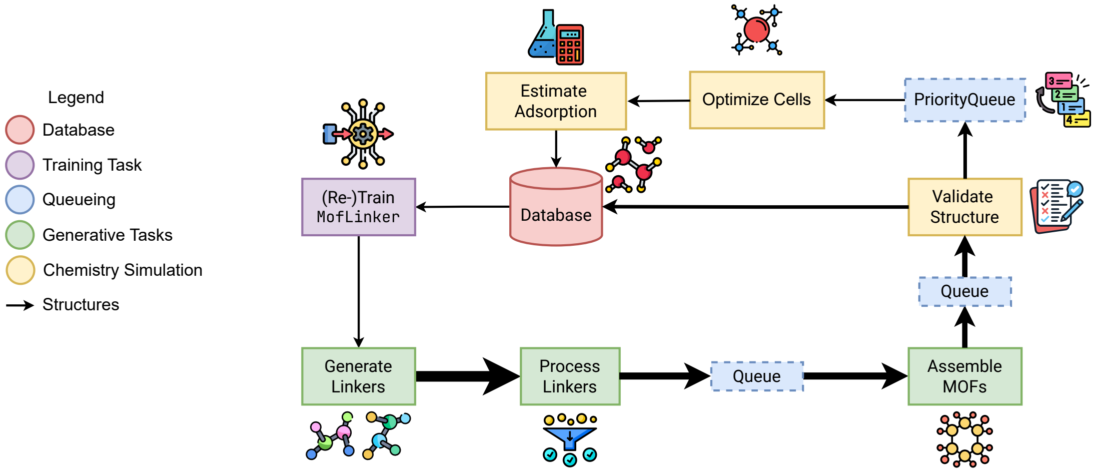

Fig. 1: `MOFA` implements an online learning loop that refines a generative AI model, `MOFLinker`, using the MOFs it has generated. The initial steps in the workflow validate linker molecules produced by the generative model before using those that pass validation to assemble MOFs. New MOFs are placed in a LIFO queue, from which they are retrieved to be evaluated for stability, and the gas capacity of the most stable are further evaluated to refine the structures and estimate properties of interest. The structures and their computed properties are collected in a database and used to retrain the GenAI model. All steps run concurrently. Note: The width of the arrows for “Structures” corresponds with the amount of structures being passed between each pair of tasks in the workflow.

dependencies between tasks are often represented as a directed acyclic graph (DAG) such that the workflow system can optimize placement and execution of tasks in the graph across local or remote resources. This programming model supports the development of sophisticated computational science applications, and thus, many workflow systems have been developed to meet the needs of the scientific community. Dask [37], FireWorks [38], Parsl [16], Pegasus [39], and Swift [40] all provide mechanisms to express a workflow (tasks and their dependencies) and a runtime for scheduling and dispatching tasks across available resources, such as an HPC cluster.

Solutions for executing an application across heterogeneous resources depend on the physical configuration of those resources. For example, the challenges faced are different when employing two distinct machines with heterogeneous hardware configurations than when utilizing a single node or a homogeneous cluster. In the multi-system case, data transfer between remote machines can limit performance and scalability. Additionally, specific tasks may be better suited for a particular hardware configuration, causing bottlenecks in throughput. In the single-system case, resource allocation and contention across heterogeneous resources (e.g., CPUs and GPUs) of single node must be considered. Some workflow systems, such as Parsl, Ray [41], and TaskVine [42], support fine-grain allocation of resources such that a subset of CPU cores, memory, or accelerators within a single node can be assigned to a task. Function-as-a-Service platforms, such as AWS Lambda [43], Google Cloud Functions [44], and Globus Compute [45], support the remote execution of tasks across but lack the fine-grain resource scheduling of workflow systems.

Many science applications have leveraged this increasing hardware heterogeneity and fine-grained workflow systems to improve their computational systems. Workflow management systems have allowed scientists to achieve new computational scales [46], [47] across a variety of fields, including virology [48], materials science [49], and astronomy [50].

III. MOFA DESIGN

Here we provide an abstract formulation for computational MOF design, detail the sequential creation of MOFs from AI generated linkers, and discuss the policies within `MOFA` that enable dynamic MOF generation.

A. Abstract Formulation

We design `MOFA` as a multi-objective, inverse material design workflow [51]–[54]. The goal of multi-objective, inverse material design is to identify materials specified by certain input variables $\mathcal{I}_i$ with properties P_j that satisfy constraints C_k. In `MOFA`, the materials being designed are MOFs, the input variables are linkers and inorganic metals, and the properties and associated constraints are defined as such to produce chemically stable MOFs with high CO_2 adsorption capacity for carbon capture.

MOF properties of MOFs can only be estimated via physical or computational experiments (i.e., there are no formal functional representations for mapping MOFs to desirable properties). For `MOFA`, we employ computational methods to estimate properties of interest. Because these properties can only be estimated experimentally, many existing and effective multi-objective optimization methods cannot be directly applied to this problem. Instead, this problem can be viewed as an *Optimal Experimental Design* (OED) problem [55] where we choose actions to perform when aiming to optimize an objective (or set of properties). In OED, it is desirable to account for the cost of different actions (e.g., time to perform certain experiments) such as by screening undesirable inputs. As an example, if given a candidate MOF, it might not be worth performing computationally-intensive estimation of properties if we can more cheaply determine that the candidate MOF is not chemically valid.

We first describe the sequential actions taken—referred to as tasks—to generate a MOF, extending the nomenclature from prior work in steering computational campaigns [17]. A **generator** $\mathcal{G}$ produces a set of linkers $l \in \mathcal{L}$ (e.g., by sampling from a known dataset or generated from an AI model). Each linker l is **screened** using an **assay** $a \in \mathcal{A}$ to estimate a **property** $P(l) \in \mathcal{P}$ of the linker with the linker being discarded if $P(l)$ does not meet some constraint. A new MOF $m \in \mathcal{M}$ is **assembled** from a subset of linkers $\mathcal{L}' \subset \mathcal{L}$. A series of increasingly expensive but discerning screening steps are applied to each MOF to find a subset of $\mathcal{M}$ with desirable properties. The specifics of this formal definition are described in Section III-B.

While each step in this sequential process could be parallelized via a single program, multiple data (SPMD) without expression of task dependencies—such an architecture would lead to inefficiencies in staging and sequencing. Instead, we choose a task parallelism architecture based on the expression of task dependencies. This allows concurrent execution of multiple actions of the same kind, and/or actions of different kinds. We describe in Section III-C how MOFA defines **policies** to determine when generation, assembly, and screening should be performed.

B. Sequential MOF Generation

The following list outlines the discovery pathway of AI-generated MOFs (see Fig. 1), after which we discuss the details of each step (summarized in Table I).

1) **Generate linkers**: Use AI model to generate linkers in a form suitable for assembly with pre-selected metal nodes.
2) **Process linkers**: Filter linkers without net-zero charge or valid valence number; prepare remainder for assembly.
3) **Assemble MOFs**: Combine linkers with metal nodes; discard if inter-atomic separations below threshold.
4) **Validate structure**: Validate MOFs for chemical soundness; compute properties; discard if below thresholds.
5) **Optimize cells**: Further optimize each MOF structure; calculate atomic partial charges of selected MOFs.
6) **Estimate adsorption**: Estimate CO_2 adsorption capacity of successful MOFs and store in database.
7) **Retrain**: Retrain MOFLinker using original linker database and linkers of newly screened MOFs.

A key challenge in generating novel molecular structures is ensuring that the model is unaffected by transformations on the molecular structure in the E(3) group (e.g., translation, reflection, rotation, and permutations) [56], [57]. DiffLinker [32] is a state-of-the-art E(3)-equivariant diffusion model originally trained to produce novel molecular structures for drug discovery. DiffLinker was trained on the GEOM dataset [58] for drug discovery; here, we fine-tune it with molecular fragments from the hypothetical MOF (hMOF) dataset [18] to produce a new model, MOFLinker, that we use to **generate MOF linkers**.

Since MOFLinker does not consider hydrogen atoms during generation (it treats them implicitly), to **process linkers** for assembly we add hydrogen atoms at appropriate locations along the linker and check that their bond lengths and angles are reasonable. OpenBabel [59] is used to determine the bond order and hydrogen atom numbers. Once hydrogen atoms are added, bond order is determined. We use the force field MMFF [60] in the RDKit [61] package to reduce stress in the linker molecule through energy minimization. Some linkers may fail these processes and are discarded; remaining linkers are a well-defined molecule with net-zero charge and valid valence number. Last, the linker anchor parts must be modified before assembly. Two types of linker are generated in the workflow: benzenecarboxylic acid (BCA) linker and benzonitrile (BZN) linker. A BCA linker's carboxylic acid groups are removed, and a dummy atom with element astatine (At) takes the carbon atom's original position; in a BZN linker, the nitrogen atoms within its cyano groups are identified, and a dummy atom with element francium (Fr) is placed 2Å away from each nitrogen atom in the direction away from the linker molecule. At and Fr are used to label dummy element sites because they are both radioactive and rarely seen in MOFs.

Thereafter, **MOF assembly** uses the processed linkers and pre-selected metal nodes to construct new MOFs. The MOF topology code label is adapted from the Reticular Chemistry Structure Resource (RCSR) database [62], a process that is automated with custom Python code. We run several assessments and preparatory tasks using RDKit to ensure that a generated MOF is both reasonable and ready to be simulated with molecular dynamics. We impute bonds for its given atomic coordinate structure, and determine its SMILES string. Then, we check that the generated MOF has reasonable bond lengths and angles. Last, we run a distance-based assessment to ensure that no pair of atoms are overlapping based on a predetermined threshold computed from the experimental database OChemDb [63]. If each of these heuristic-based steps pass, the MOF is ready for simulation; if not, it is discarded.

The next step is to **validate structures** of MOFs with molecular dynamics simulations. A pre-simulation screen, cif2lammps [64], is used to ensure that atomic structures and chemical bonds are chemically valid within the scope of UFF4MOF [65], [66], a force field used to accelerate the optimization of MOF structures. Then, a LAMMPS [14] simulation is performed to examine the stability and porous properties of MOFs that have passed prior screens. For each MOF, a 2×2×2 supercell structure is equilibrated under a triclinic isothermal-isobaric ensemble at $\langle p \rangle = 1$ atm and $\langle T \rangle = 300$K, such that the cell lengths and angles of the MOF structure can be equilibrated. These simulations are run for 10^6 steps with a step size of 0.5 fs. The Linear Lagrangian Strain Tensor (LLST): $S = 0.5(e + e^T)$ is calculated for each MOF, where $e = R_2 R_1^{-1} - I$; R_1 and R_2 are the unit cell vectors for the initial MOF structure and the final MOF structure after

TABLE I: Details for task types described in Section III-B. Some tasks employ multiple steps or codes. *Remain* is the percent of the original structures (linkers for the first two tasks, MOFs for subsequent tasks) that remain, on average, after the task is performed. *Time* in the last column is per structure except for retraining, which is the time to re-train over the entire dataset. The resource, remain, and time values are those chosen for or observed during our 450 node run.

Task	**Type**	**Description**	**Code**	**Resource**	**Remain (%)**	**Time (s)**
Generate linkers	Generate	Generate novel linkers	MOFLinker/PyTorch	1 GPU	100.0	0.37
Process linkers	Screen	Screen/optimize linkers	RDKit/OpenBabel	1 CPU	22.8	0.12
Assemble MOFs	Assemble	Connect linkers & metal clusters	Custom	1 CPU	100.00	0.46
	Screen	Check bonds & atomic distances	RDKit	1 CPU	99.90	2.56
Validate structure	Screen	Check geometry & bonds	cif2lammps	0.5 GPU	15.20	19.98
	Screen	Test stability & porosity	LAMMPS	0.5 GPU	8.60	204.52
Optimize cells	Screen	Optimize cell structure	CP2K	2 nodes	0.03	1517.53
Estimate adsorption	Screen	Compute partial charges	Chargemol	1 CPU	0.03	211.78
	Estimate	Estimate CO_2 adsorption	RASPA	1 CPU	0.03	1892.89
Retrain	Retrain	Retrain with newly screened MOFs	MOFLinker/PyTorch	1 node		96.50

the LAMMPS simulation; and I is the 3×3 identity matrix. Eigenvalues of the LLST are calculated, and the maximum absolute value of these eigenvalues is chosen as the metric to evaluate the lattice distortion before and after the simulation.

CP2K v2024.1 with Quickstep [13], [67] is then used to **optimize cells** for each MOF. Each calculation starts with an initial structure from the prior molecular dynamics simulations, which is then optimized with a limited number of L-BFGS [68] steps. Gaussian and Plane Wave (GPW) method along with Perdew–Burke–Ernzerhof (PBE) [69] exchange-correlation functional is applied. The short range variant of the molecularly optimized basis functions with double-zeta valence recommended by Goedecker, Teter, and Hutter (DZVP-MOLOPT-SR-GTH) [70], [71] are used. Additionally, DFT-D3 of van der Waals correction by Grimme [72] is added.

The atomic partial charge is calculated using the Chargemol program with the Density Derived Electrostatic and Chemical (DDEC6) method [73], [74]. MOFs electronic density in the 3D space is calculated by a single-point energy calculation with CP2K, and the Chargemol program estimates the point charge on each atom that would best fit the calculated electronic density. The MOFs failed in atomic partial charge assignment are discarded. If the MOFs are successfully assigned with atomic partial charge, their CO_2 adsorption value are evaluated using the Grand Canonical Monte Carlo (GCMC) simulation in RASPA [15] (i.e., **estimate adsorption**). Specifically, we want to estimate CO_2 capacity at 0.1 bar pressure and 300 K. Given the high computational cost and serial execution of GCMC, simulations are conducted under the assumption that MOF structures are rigid. The atoms of the MOF structures are assigned with Lennard-Jones parameters from the UFF4MOF force field; the default force field model for CO_2 within RASPA is used. Coulomb forces capture electrostatic interactions in MOFs, crucial for gas adsorption. Ewald summation efficiently handles long-range interactions in periodic systems. Together, they enhance GCMC, enabling accurate estimates of CO_2 capacity in MOFs. Adsorption capacities are stored in the database.

Periodically, MOFLinker is **retrained** on MOFs identified by previous computations. Retraining starts from the weights learned from pre-training on the hMOF and GEOM datasets [58], and uses a new training set of linkers from as few as 32 and as many as 8192 of the best-performing MOFs yet found during a run. The training sets are composed of MOFs with high stability (<25% lattice strain) and, at first, only those in the lowest 50% of lattice strain and then, after 64 gas adsorption calculations have completed, only those with the highest gas adsorption. Our intent is for the fine-tuned MOFLinker models to generate linkers similar to those in MOFs with optimal stability and capacity. Retraining is first performed once 64 stability calculations have completed, and subsequently after the preceding retraining run has finished and the training set size expands by any amount. Retraining requires 30–300 seconds, depending on training set size.

C. Workflow Policies

Policies are necessary to dynamically determine what steps to perform at any moment because the sequential screening of entities (linkers and MOFs) means that the number of possible actions and the cost of each action varies throughout the execution of MOFA. MOFA utilizes the following policies:

- Linkers are continuously generated and processed.
- MOF assembly is performed on the most recently generated linkers as soon as enough linkers four linkers of each type (BCA and BZN) are available. Assembly runs continuously on one parallel worker for every 256 used for stability calculations.
- Computations are performed to assess stability of the most recently assembled MOFs, with a new computation started whenever a stability worker is idle.
- Adsorption computations are performed on the most stable MOFs, again with sufficient computations running to maintain full utilization of available workers.

- MOFLinker is retrained when at least 64 MOFs with lattice strains below 25% have been found.

The concurrent execution of many steps, with information flowing from one to another and a need to access the most recent (or, in the case of adsorption calculations, the most stable) entity in order to maximize scientific performance, introduces many execution challenges which we discuss in the following section.

IV. Executing MOFA

The dynamic mix of tasks within MOFA requires careful attention to policy expression, scheduling, and resource allocation to achieve both high system utilization and efficient and scalable MOF discovery. We architect MOFA to leverage heterogeneous resources and to reduce system latencies (e.g., time to receive a task result) so that new tasks can be determined based on up-to-date information.

A. Policy Expression

We build MOFA on Colmena [17], [75], [76], a Python library for steering simulation ensembles. In Colmena, a central process, the *Thinker*, executes a set of policies expressed through *agents* that can perform actions by submitting *tasks* to a *Task Server*, which manages the remote, asynchronous execution of tasks requested by agents. Functionally, a task is implemented as a Python function that is executed on a remote process, and agents are threads within the main Thinker process that manage resources, submit tasks, and process task results.

Each of the seven steps described in Section III-B are tasks managed by Colmena agents. A set of agents are implemented to express the MOFA policies described in Section III-C. For example, one agent is responsible for receiving assembled MOFs, notifying a second agent that resources are available for a new **assemble MOFs** task, and adding the new MOF to a LIFO queue, to be processed by a third agent that launches a **validate structures** task when resources are available. The goal of these policies is to ensure that resources are appropriately allocated between tasks, such as to avoid allocating resources to validate structures when there are insufficient assembled MOFs, and to ensure timely propagation between tasks so that agents make decisions with the most up-to-date information, such as by allocating resources for simulating a more recently created MOF (assuming that MOF quality improves over time).

Prior to this work, Colmena did not have a way to express generator tasks—i.e., tasks that continually yield intermediate data without necessarily returning—which makes it challenging to express MOFA's generative AI tasks. Thus, we extended Colmena to support Python generator functions that stream intermediate results back to a central process to be consumed and acted upon by an agent.

B. Resource Allocation and Communication

MOFA uses Parsl to schedule and execute tasks. We configure a Parsl executor for each resource type and map task types within Colmena to the respective executors. Rather than submitting a large bag-of-tasks to Parsl, MOFA only submits tasks when resources allocated to a task type are available. This choice enables agents to reallocate resources amongst task types depending on the dynamic load across workflow components (e.g., queue lengths). Notification of a task completion may trigger reallocation of resources and must be done swiftly to maintain high utilization of those resources. Realizing responsive communication requires reducing costs associated with transmitting results from compute workers to the Thinker and for the Thinker to use them to then decide the next task.

We optimize communication latency by separating workflow control messages (e.g., those used by Colmena and Parsl) from result data transfer with ProxyStore [77], [78]. Sending data through a separate channel speeds the workflow engine's control process, and decouples actions that involve simply knowing that a task has completed from those that require reading the data. For example, the Thinker launches the next atomistic simulation as soon as another finishes ($O(1)$ ms latency) and then launches a retraining task once the data from the simulation is processed ($O(100)$ ms latency).

We further accelerate the decision process by distributing the compute-intensive parts of the post-processing (e.g., **process linkers**) across idle cores on compute nodes. Distributing tasks to idle cores prevents agents in the Thinker from having to perform post-processing themselves—which would otherwise slow down its ability to respond to new events as quickly. A final strategy is to process batches of results from inference tasks while others are being run. We stream results from inference workers to idle cores on other nodes.

Using our Colmena agents and Parsl executors, we allocate resources for task type as follows (see Table I):

1) **Generate linkers** is performed on a single GPU with a batch size selected to maximize GPU utilization.
2) **Process linkers**, **assemble MOFs**, and **estimate adsorption** tasks are placed on the idle cores of nodes running **validate structure** tasks. All tasks are isolated by enforcing thread affinity.
3) **Validate structure** is configured such that two task invocations share one GPU (via NVIDIA's Multi-Process Service, MPS [79]) but are pinned to different CPUs.
4) **Optimize cells** runs across two dedicated nodes via MPI.
5) **Retrain** is performed in a data parallel fashion across all 4 GPUs of a single dedicated node.

The agents cooperate to re-allocate available resources from the pool between tasks types as needed. A visualization of these and how they are executed across nodes in the HPC cluster can be seen in Fig. 2.

V. Evaluation

We measure MOFA performance along two axes: *(i)* computational efficiency on an HPC cluster and *(ii)* scientific output. We performed experiments on between 32 and 450 nodes of the Argonne Leadership Computing Facility's Polaris Supercomputer, an HPE Apollo supercomputer with 560 nodes, each with one AMD EPYC Milan 7543P (32-core, 2.8 GHz) and four 40 GB NVIDIA A100 GPUs.

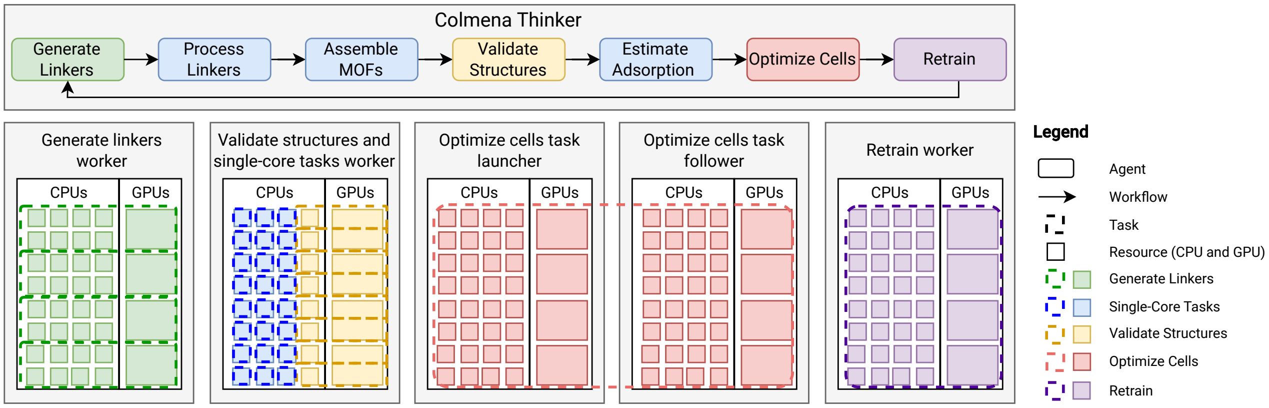

Fig. 2: Task and resource allocation in the MOFA workflow. The top section shows the Colmena Thinker, containing seven agents (rounded-corner boxes), each corresponding to one of the seven tasks. The bottom section depicts five types of MOFA workers, each with a 32-core CPU and four GPUs, with distinct resource allocation schemata for different MOFA tasks.

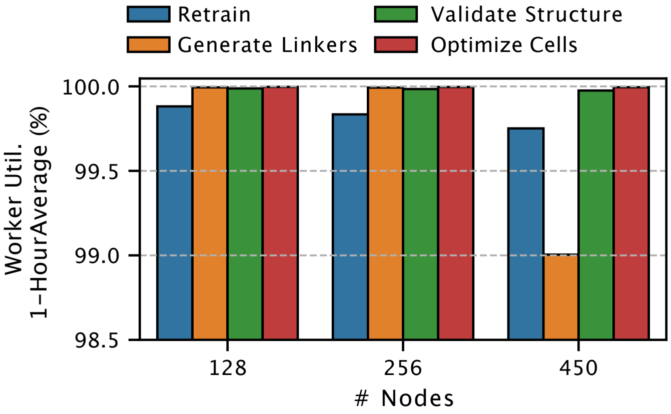

Fig. 3: Active time of compute nodes on Polaris, as measured by the average time each workflow worker spent processing work over one hour.

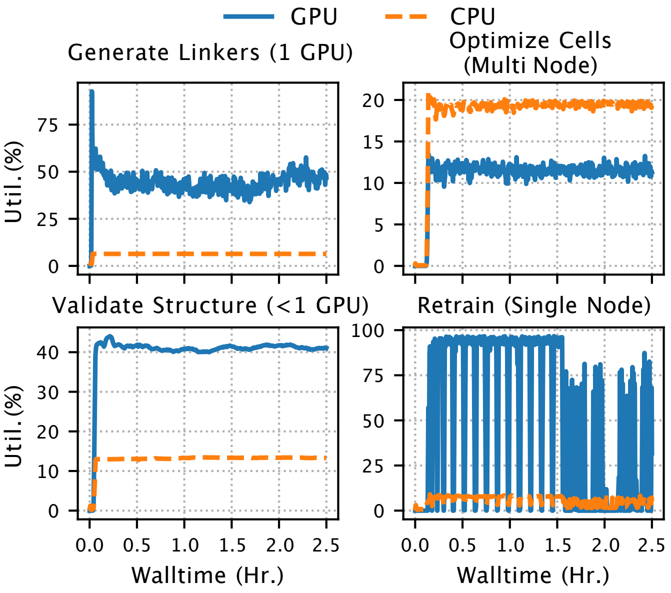

Fig. 4: MOFA's utilization of Polaris compute nodes as fraction of peak varies with the code running.

A. *Utilization of Heterogeneous Resources*

We first calculate the fraction of time that workers spend doing useful work by analyzing timestamps generated when each worker starts and completes a task. As shown in Fig. 3, the workers for all four task types spend over 99% of their time executing tasks. The workflow makes effective use of every one of the 450 nodes.

Next, we inspect the utilization of hardware within the nodes. MOFA achieves consistent utilization across each type of node during the entire 3-hour run. We see in Fig. 4 that average GPU and CPU utilization remain constant during a 450-node run for all except the single node workers. The single node workers run training tasks, which are large and frequent during the beginning of the run when the application is retraining on any stable MOF and infrequent as the training waits until new gas capacity computations complete. Only the single node workers maintain near-100% utilization of the GPU and all have less than full utilization of the CPU. This suggests that we can benefit from further use of NVIDIA's Multi Process Service across all other worker types that under-utilize the GPU (i.e., all but single node workers) in order to execute more tasks per node.

There is room available to offload more post-processing from the Thinker to idle CPUs on the compute nodes. The validate structure tasks, which are allocated <1 GPU, use approximately one quarter of the CPU cores throughout the entire run, and generate linkers tasks, which are allocated one GPU, use only one eighth. Thus, there remain idle CPUs and it is possible to continue our strategy of distributed post-processing across idle cores without hindering other tasks.

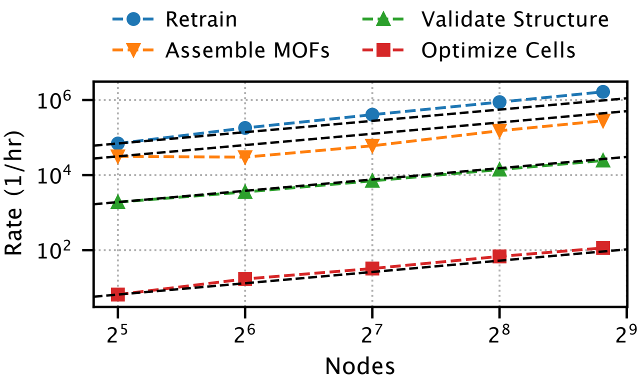

Fig. 5: Sustained throughput in tasks per hour for the four main workflow stages as a function of system scale. The dashed lines indicate ideal scaling computed from the rates at the smallest node count.

B. *Effect of Scale on Task Throughput*

The rate at which our application evaluates new MOFs increases linearly with scale, as desired. We measured the throughput by counting the total number of generated linkers, assembled MOFs, structures validated, and cells optimized and estimated then determining a sustained rate using linear regression. As shown in Fig. 5, the throughput for each stage increases linearly from 32 nodes up to a full machine run.

The key to scaling in MOFA is low inter-stage latencies in the pipeline, because the MOFA steering logic only submits enough tasks as available compute resources: results must be processed and new tasks submitted without delay to allow later stages to work on the most up-to-date data. Consequently, we assess the crucial timings between several stages of the execution plan as a function of scale to identify potential bottlenecks to further scaling. As shown in Fig. 6, we find that the latencies for each of the five critical steps of our application are not degraded by scale. We assess each below:

- Process linkers latency is the time between generating a batch of linkers in a generate linkers task to the Thinker receiving the processed batch from a process linkers task. This $O(10)$ s latency is primarily due to the process linkers task runtime; it is constant across node counts, indicating that sufficient CPUs are available for processing. It could be reduced by increasing the parallelism of batch processing.
- Validate structures latency is the time between a LAMMPS simulation completing and its result being stored in the database.
- Retrain latency is the time between finishing retraining a model to that model being used in a generate linkers task. Generate linkers tasks complete more frequently at larger scales, leading to a lower latency with scale. The latency could be further reduced by adding a mechanism to halt generate linkers tasks when a new model is available.
- Compute partial charges latency is the time from an optimize cells task finishing to an estimate adsorption task starting. It remains ∼1 s at all scales.
- Estimate adsorption latency is the time between screening and estimation within estimate adsorption tasks. This also reaches ∼1 s at the largest scale.

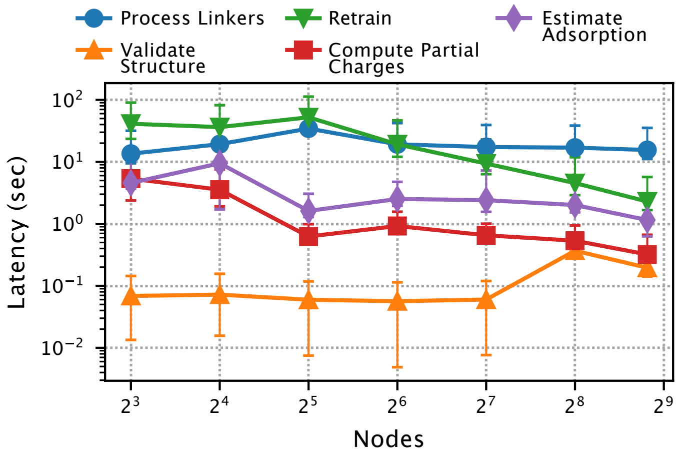

Fig. 6: Mean and inter-quartile range of key latencies, defined in Section V-B, as a function of node count.

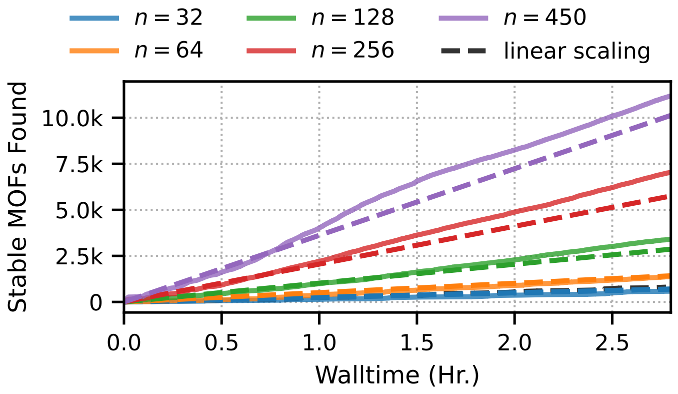

Fig. 7: Number of stable MOFs found over time for MOFA runs on from 32 to 450 nodes. Dashed lines indicated the number of stable MOFs expected by scaling the rate of finding stable MOFs over the entire duration of the 32-node run.

Latencies are kept low by high speed interconnects and messages that for many steps are much less than 1 MB, which do not saturate the network. The largest tasks (assemble MOFs with 10–40 MB inputs and 1–2 MB outputs, process linkers with 100–500 KB inputs and outputs, and validate structures with 400–600 KB outputs) do require high performance connections to keep communication time in the sub-second range. We observe >1 GB/s transfer rates for many assemble MOF tasks, in particular. These bandwidth requirements are clearly achievable for Polaris at near full-system scales, and we do not anticipate problems with further scaling.

C. Ability to Find Stable MOFs

Fig. 7 shows the number of stable MOFs (defined as LAMMPS calculation indicating $<10\%$ chemical strain) over time. We attribute the modest increase over time in the rate at which stable MOFs are generated to repeated retraining of DiffLinker improving the quality of generated linkers.

We assessed the effect of retraining by repeating the 32-node and 64-node runs with the retraining portion of the workflow disabled. The effect of retraining on the stable MOF discovery rate is significant, increasing the number found at 90 minutes from 133 to 313 on 32 nodes and from 393 to 641 on 64 nodes. The increase in performance is because the fraction of MOFs found to be stable improves. The fraction of MOFs found to be stable increases from 5 to 11% when using retraining on 32 nodes and from 8 to 12% for 64 nodes. Learning from intermediate workflow results is clearly beneficial in MOFA.

The resources devoted to retraining stay constant at different scales, yet the impact becomes larger. After 90 minutes, the 450-node run has found 9.7 stable MOFs per node hour expended, vs. 9.5 for the 256-node run and only 6.5 for the 32-node. We attribute the steady improvement in discovery rate prior to 90 minutes for the 450-node case to more data being gathered, leading to better machine learning models, and—consequently—a more effective MOFLinker at the same walltime. (The rate for the 450-node run diminishes after 90 minutes because, unlike the smaller runs, MOFA has by then acquired enough data to switch from retraining based on only stability to a a more stringent combination of stability and gas adsorption capacity.)

D. Novelty and Chemistry Insights of Generated MOFs

To evaluate the effectiveness of using MOFA to search for MOFs with high stability and CO_2 adsorption, we evaluated the molecules chemical properties over time. In Fig. 10 we compared the cumulative distribution functions (CDFs) of chemical stability of the generated MOFs for each hour MOFA ran. We observe that over time the stability of the MOFs increased, shown by a larger proportion of MOFs having a lower chemical strain. This suggests that our MOFA workflow is properly learning to generate MOFs for one of our target objectives. To understand the chemical novelty of these MOFA-generated molecules, in Fig. 9 we plot embedding representations of each of the molecules using a UMAP projection based on 38 chemical properties. While some areas of chemical space were shared between the hMOF database and the MOFA-generated linkers, we find that our approach provides candidates that are chemically diverse while sharing important chemical similarities with previously identified successful MOFs.

While the chemical stability of the generated MOFs was promising, the goal of MOFA is to generate molecules with high CO_2 adsorption. The MOFA-generated MOFs include one with CO_2 capacity in the top five of the hMOF dataset, i.e., 4.05 mol/kg at 0.1 bar: see Fig. 8. Ten other MOFs produced by the 450-node run also rank in the top 10% of hMOF, with capacities of 1–2 mol/kg at 0.1 bar. In brief, with a single 450-node, 3-hour run, MOFA has enabled us to build a set of novel MOFs with good, and one with very high, CO_2 capacities. These results demonstrate MOFA's capabilities for materials science discovery, and suggest avenues to further improve its performance.

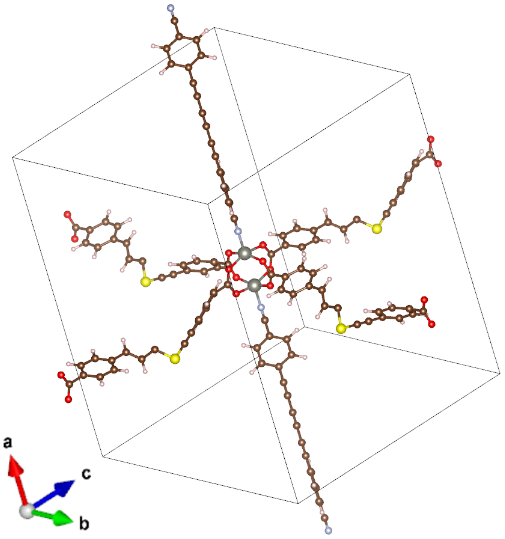

Fig. 8: The generated MOF with highest CO_2 capacity (4.05 mol/kg at 0.1 bar) produced by a 450-node, 3-hour MOFA run on Polaris. Brown: carbon; red: oxygen; white: hydrogen; yellow: sulfur; grey (big): zinc; blue white (small): nitrogen.

VI. Impact & Future Work

MOFA, as a high-throughput generative workflow, presents an opportunity for the discovery of novel MOFs that have a wide array of applications (e.g., reducing greenhouse gas emissions, catalysis); it is also a compelling application for AI and systems researchers interested in active learning, task scheduling, and data management at large scales.

A. Efficient MOF Discovery

The promise of MOFs for carbon capture is multifaceted: *(i)* their high surface area (from 1000 to 10,000 m^2g^{-1}, exceeding those of traditional porous materials such as carbons and zeolites) and porosity improve CO_2 adsorption and selectivity [80]; *(ii)* their pore size and shape may be tailored by carefully selecting their organic linkers and the connectivity of the metal ion clusters; *(iii)* they may be designed to maintain their structural integrity under harsh environmental conditions; and *(iv)* they can be fabricated at large scale with low-cost and simple synthetic methods [81]. All these features promote MOFs as a desirable future energy material for carbon capture [82], [83]. Furthermore, given their unique properties, MOFs have been applied in a number of areas beside carbon capture, including energy storage, catalysis, optoelectronics, and sensing [84]. Researchers could therefore adjust the simulation criteria of MOFA to search for novel MOFs with applications beyond carbon capture.

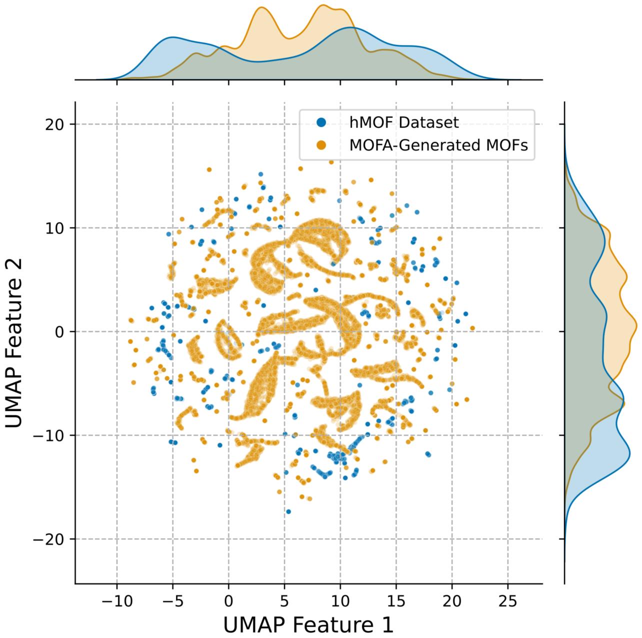

Fig. 9: UMAP plot of the diversity of MOFA-generated linkers compared to linkers from the hMOF database (represented with an RDKit embedding). While some regions of chemical space overlap between hMOF and MOFA-generated linkers, the latter explores structures and moieties that differ significantly from those in the original training set—highlighting MOFA's ability to discover new structures within the space of hMOF.

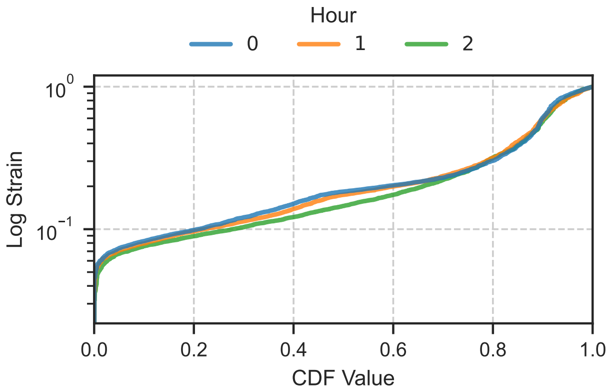

Fig. 10: The empirical cumulative distribution of the stability of MOFs generated by our 64-node run binned by the hour they were generated. MOF stability (measured by strain) improves over time as the workflow runs.

However, the many possible clusters and linkers mean that, in principle at least, millions of different MOFs may be created with different properties simply by varying the choice of building blocks. Experimental screening of millions of potential MOFs is impractical, and atomistic simulations, like experiments, are too expensive to be used for trial-and-error exploration of the vast MOF chemical design space. MOFA presents a rigourous approach for efficiently exploring this space by coupling GenAI, high-throughput screening, and atomistic simulations to accelerate the rational discovery of stable, chemically diverse, and high performing MOFs.

We note several other ways in which MOFA can be applied for scientific discovery. We have applied it to create an open source database of high-quality MOFs (URL ommitted for double-blind review). We aspire also to connect MOFA with robotics laboratories that synthesize high performing MOFs, and then provide input data regarding experimental synthesizability scores, and cost-effectiveness to manufacture and use such MOFs at scale.

B. Algorithm Research Opportunities

MOFA also presents opportunities for algorithm research. As described in Fig. 1, MOFA employs queue prioritization strategies to determine which structures are used by the next stage in the workflow. However, many molecule screening procedures (e.g., those in the context of *de novo* drug discovery [85]) take advantage of adaptive approaches that modify the experiments performed during screening based on new information. MOFA can be readily configured to permit adaptive approaches such as active learning or model-predictive control: for example, by dynamically re-prioritizing queues with an active learning agent that optimizes different workflow objectives (e.g., candidate stability, diversity, gas capacity). Better algorithms can improve scientific outcomes and/or improve resource allocation, such as by re-prioritizing the DFT simulation queue so that computationally expensive experiments are only performed on structures with high predicted gas capacity.

C. Systems Research Opportunities

MOFA's modular design facilitates the evaluation of different technologies. The configuration of Section IV works well for our execution environment, but can easily be adapted to support future systems research. The MOFA workflow also represents a unique workload due to its complexity and heterogeneity.

The Colmena system that MOFA uses for orchestration exposes abstractions including Thinker to Task Server queues for transmitting task requests and streaming results; ProxyStore for intermediate data transfer; and the Task Server for task execution. Each of these abstractions enables the use of alternate implementations. For example, researchers can evaluate message broker systems by comparing task latencies in MOFA. ProxyStore enables comparing scalability, latency, and throughput of object stores for intermediate data transfer with its robust plugin system. Alternate Task Server implementations can be easily created to execute MOFA with different execution engines/workflow systems.

VII. Conclusion

We have presented MOFA, an HPC-coupled-generative AI workflow for the accelerated discovery of MOFs for carbon capture. This workflow leverages heterogeneous computing resources to maximize novel MOF generation through the orchestration of generative-AI, high throughput in-silica screening, and high fidelity atomistic simulations. We optimized

MOFA's performance by running tasks asynchronously across these workflow modules to maximize resource utilization and throughput of stable MOF generation. Once generated, these MOFs were screened for stability and CO$_2$ capacity. MOFA is capable of generating over 100 novel MOFs per hour, and produced 11 promising new candidates for carbon capture in a 450-node three hour run. We demonstrate the effectiveness of MOFA in novel MOF design for carbon capture, while also highlighting the modular nature of our workflow. It is our hope that MOFA's modular design will enable future research efforts in distributed systems, as well as in materials science and other science domains that involve AI and large scale simulations.

Acknowledgments

This work was supported by Laboratory Directed Research and Development (LDRD) funding from Argonne National Laboratory, provided by the Director, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-06CH11357. This research was partially supported by the Catalyst Design for Decarbonization Center, an Energy Frontier Research Center funded by the U.S. Department of Energy, Office of Science, Basic Energy Sciences under award no. DE-SC0023383. This work used resources of the Argonne Leadership Computing Facility, a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.

References

[1] J. Hansen, D. Johnson, A. Lacis, S. Lebedeff, P. Lee, D. Rind, and G. Russell, "Climate impact of increasing atmospheric carbon dioxide," *Science*, vol. 213, pp. 957–966, Aug. 1981.

[2] Y. Zhang, Y. Zhang, X. Wang, J. Yu, and B. Ding, "Ultrahigh metal–organic framework loading and flexible nanofibrous membranes for efficient CO2 capture with long-term, ultrastable recyclability," *ACS Applied Materials & Interfaces*, vol. 10, no. 40, pp. 34802–34810, 2018.

[3] M. Fernandez, P. G. Boyd, T. D. Daff, M. Z. Aghaji, and T. K. Woo, "Rapid and accurate machine learning recognition of high performing metal organic frameworks for CO2 capture," *The Journal of Physical Chemistry Letters*, vol. 5, no. 17, pp. 3056–3060, 2014.

[4] H. Li, K. Wang, Y. Sun, C. T. Lollar, J. Li, and H.-C. Zhou, "Recent advances in gas storage and separation using metal–organic frameworks," *Materials Today*, vol. 21, no. 2, pp. 108–121, 2018.

[5] M. Hao, M. Qiu, H. Yang, B. Hu, and X. Wang, "Recent advances on preparation and environmental applications of MOF-derived carbons in catalysis," *Science of the Total Environment*, vol. 760, p. 143333, 2021.

[6] H. D. Lawson, S. P. Walton, and C. Chan, "Metal–organic frameworks for drug delivery: A design perspective," *ACS Applied Materials & Interfaces*, vol. 13, no. 6, pp. 7004–7020, 2021.

[7] Y. Zhang, S. Yuan, G. Day, X. Wang, X. Yang, and H.-C. Zhou, "Luminescent sensors based on metal-organic frameworks," *Coordination Chemistry Reviews*, vol. 354, 08 2017.

[8] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

[9] K. M. Jablonka, D. Ongari, S. M. Moosavi, and B. Smit, "Big-data science in porous materials: Materials genomics and machine learning," *Chemical Reviews*, vol. 120, no. 16, pp. 8066–8129, 2020.

[10] Y. Kang and J. Kim, "ChatMOF: An artificial intelligence system for predicting and generating metal-organic frameworks using large language models," *Nature Communications*, vol. 15, no. 1, p. 4705, 2024.

[11] H. Park, X. Yan, R. Zhu, E. A. Huerta, S. Chaudhuri, D. Cooper, I. Foster, and E. Tajkhorshid, "A generative artificial intelligence framework based on a molecular diffusion model for the design of metal-organic frameworks for carbon capture," *Communications Chemistry*, vol. 7, no. 1, p. 21, 2024.

[12] S. M. Moosavi, A. Nandy, K. M. Jablonka, D. Ongari, J. P. Janet, P. G. Boyd, Y. Lee, B. Smit, and H. J. Kulik, "Understanding the diversity of the metal-organic framework ecosystem," *Nature Communications*, vol. 11, no. 1, pp. 1–10, 2020.

[13] T. D. Kühne, M. Iannuzzi, M. D. Ben, V. V. Rybkin, P. Seewald, F. Stein, T. Laino, R. Z. Khaliullin, O. Schütt, F. Schiffmann, D. Golze, J. Wilhelm, S. Chulkov, M. H. Bani-Hashemian, V. Weber, U. Borštnik, M. Taillefumier, A. S. Jakobovits, A. Lazzaro, H. Pabst, T. Müller, R. Schade, M. Guidon, S. Andermatt, N. Holmberg, G. K. Schenter, A. Hehn, A. Bussy, F. Belleflamme, G. Tabacchi, A. Glöß, M. Lass, I. Bethune, C. J. Mundy, C. Plessl, M. Watkins, J. VandeVondele, M. Krack, and J. Hutter, "CP2K: An electronic structure and molecular dynamics software package - Quickstep: Efficient and accurate electronic structure calculations," *The Journal of Chemical Physics*, vol. 152, p. 194103, May 2020.

[14] A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in 't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott, and S. J. Plimpton, "LAMMPS - A flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales," *Computer Physics Communications*, vol. 271, p. 108171, 2022.

[15] D. Dubbeldam, S. Calero, D. E. Ellis, and R. Q. Snurr, "RASPA: Molecular simulation software for adsorption and diffusion in flexible nanoporous materials," *Molecular Simulation*, vol. 42, no. 2, pp. 81–101, 2016.

[16] Y. Babuji, A. Woodard, Z. Li, D. S. Katz, B. Clifford, R. Kumar, L. Lacinski, R. Chard, J. Wozniak, I. Foster, M. Wilde, and K. Chard, "Parsl: Pervasive Parallel Programming in Python," in *28th ACM International Symposium on High-Performance Parallel and Distributed Computing (HPDC)*, 2019.

[17] L. Ward, G. Sivaraman, J. Pauloski, Y. Babuji, R. Chard, N. Dandu, P. C. Redfern, R. S. Assary, K. Chard, L. A. Curtiss, R. Thakur, and I. Foster, "Colmena: Scalable machine-learning-based steering of ensemble simulations for high performance computing," in *IEEE/ACM Workshop on Machine Learning in High Performance Computing Environments*, (Los Alamitos, CA, USA), pp. 9–20, IEEE Computer Society, nov 2021.

[18] C. E. Wilmer, O. K. Farha, Y.-S. Bae, J. T. Hupp, and R. Q. Snurr, "Structure–property relationships of porous materials for carbon dioxide separation and capture," *Energy & Environmental Science*, vol. 5, no. 12, p. 9849, 2012.

[19] Q. Wang and D. Astruc, "State of the art and prospects in metal–organic framework (MOF)-based and MOF-derived nanocatalysis," *Chemical reviews*, vol. 120, no. 2, pp. 1438–1511, 2019.

[20] L. P. L. Mosca, A. B. Gapan, R. A. Angeles, and E. C. R. Lopez, "Stability of metal-organic frameworks: Recent advances and future trends," in *4th International Electronic Conference on Applied Sciences*, p. 146, MDPI, Nov. 2023.

[21] M. Fujita, Y. J. Kwon, S. Washizu, and K. Ogura, "Preparation, clathration ability, and catalysis of a two-dimensional square network material composed of cadmium (II) and 4, 4'-bipyridine," *Journal of the American Chemical Society*, vol. 116, no. 3, pp. 1151–1152, 1994.

[22] A. Corma, H. Garcia, and F. Llabrés i Xamena, "Engineering metal organic frameworks for heterogeneous catalysis," *Chemical Reviews*, vol. 110, no. 8, pp. 4606–4655, 2010.

[23] Q. Yang, Q. Xu, and H.-L. Jiang, "Metal–organic frameworks meet metal nanoparticles: Synergistic effect for enhanced catalysis," *Chemical Society Reviews*, vol. 46, no. 15, pp. 4774–4808, 2017.

[24] Y.-Z. Chen, R. Zhang, L. Jiao, and H.-L. Jiang, "Metal–organic framework-derived porous materials for catalysis," *Coordination Chemistry Reviews*, vol. 362, pp. 1–23, 2018.

[25] C. E. Wilmer, M. Leaf, C. Y. Lee, O. K. Farha, B. G. Hauser, J. T. Hupp, and R. Q. Snurr, "Large-scale screening of hypothetical metal–organic frameworks," *Nature Chemistry*, vol. 4, no. 2, pp. 83–89, 2012.

[26] H. Daglar and S. Keskin, "Computational screening of metal–organic frameworks for membrane-based CO2/N2/H2O separations: Best materials for flue gas separation," *The Journal of Physical Chemistry C*, vol. 122, no. 30, pp. 17347–17357, 2018.

[27] M. Fernandez and A. S. Barnard, "Geometrical properties can predict CO2 and N2 adsorption performance of metal–organic frameworks (MOFs) at low pressure," *ACS Combinatorial Science*, vol. 18, no. 5, pp. 243–252, 2016.

[28] K. Mukherjee, A. W. Dowling, and Y. J. Colón, "Sequential design of adsorption simulations in metal–organic frameworks," *Molecular Systems Design & Engineering*, vol. 7, no. 3, pp. 248–259, 2022.

[29] K. Swanson, G. Liu, D. B. Catacutan, A. Arnold, J. Zou, and J. M. Stokes, "Generative AI for designing and validating easily synthesizable and structurally novel antibiotics," *Nature Machine Intelligence*, vol. 6, no. 3, pp. 338–353, 2024.

[30] J. Li, M. Zhang, J. R. Martins, and C. Shu, "Efficient aerodynamic shape optimization with deep-learning-based geometric filtering," *AIAA Journal*, vol. 58, no. 10, pp. 4243–4259, 2020.

[31] E. Kim, K. Huang, S. Jegelka, and E. Olivetti, "Virtual screening of inorganic materials synthesis parameters with deep learning," *npj Computational Materials*, vol. 3, no. 1, p. 53, 2017.

[32] I. Igashov, H. Stärk, C. Vignac, A. Schneuing, V. G. Satorras, P. Frossard, M. Welling, M. Bronstein, and B. Correia, "Equivariant 3D-conditional diffusion models for molecular linker design," *Nature Machine Intelligence*, pp. 1–11, 2024.

[33] X. Fu, T. Xie, A. S. Rosen, T. Jaakkola, and J. Smith, "MOFDiff: Coarse-grained diffusion for metal-organic framework design," *arXiv preprint arXiv:2310.10732*, 2023.

[34] Z. Yao, B. Sánchez-Lengeling, N. S. Bobbitt, B. J. Bucior, S. G. H. Kumar, S. P. Collins, T. Burns, T. K. Woo, O. K. Farha, R. Q. Snurr, and A. Aspuru-Guzik, "Inverse design of nanoporous crystalline reticular materials with deep generative models," *Nature Machine Intelligence*, vol. 3, no. 1, pp. 76–86, 2021.

[35] G. Fox and S. Jha, "Learning everywhere: A taxonomy for the integration of machine learning and simulations," in *15th International Conference on eScience*, pp. 439–448, IEEE, 2019.

[36] N. Dey, G. Gosal, Zhiming, Chen, H. Khachane, W. Marshall, R. Pathria, M. Tom, and J. Hestness, "Cerebras-GPT: Open Compute-Optimal Language Models Trained on the Cerebras Wafer-Scale Cluster," 2023.

[37] M. Rocklin, "Dask: Parallel computation with blocked algorithms and task scheduling," in *14th Python in Science Conference*, vol. 130, p. 136, 2015.

[38] A. Jain, S. P. Ong, W. Chen, B. Medasani, X. Qu, M. Kocher, M. Brafman, G. Petretto, G.-M. Rignanese, G. Hautier, D. Gunter, and K. A. Persson, "FireWorks: A dynamic workflow system designed for high-throughput applications," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 17, pp. 5037–5059, 2015.

[39] E. Deelman, K. Vahi, G. Juve, M. Rynge, S. Callaghan, P. J. Maechling, R. Mayani, W. Chen, R. Ferreira da Silva, M. Livny, and K. Wenger, "Pegasus, a workflow management system for science automation," *Future Generation Computer Systems*, vol. 46, pp. 17–35, 2015.

[40] M. Wilde, M. Hategan, J. M. Wozniak, B. Clifford, D. S. Katz, and I. Foster, "Swift: A language for distributed parallel scripting," *Parallel Computing*, vol. 37, no. 9, pp. 633–652, 2011.

[41] P. Moritz, R. Nishihara, S. Wang, A. Tumanov, R. Liaw, E. Liang, M. Elibol, Z. Yang, W. Paul, M. I. Jordan, and I. Stoica, "Ray: A distributed framework for emerging AI applications," in *13th USENIX Conference on Operating Systems Design and Implementation*, OSDI'18, (USA), p. 561–577, USENIX Association, 2018.

[42] B. Sly-Delgado, T. S. Phung, C. Thomas, D. Simonetti, A. Hennessee, B. Tovar, and D. Thain, "TaskVine: Managing in-cluster storage for high-throughput data intensive workflows," in *SC '23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis*, SC-W '23, (New York, NY, USA), p. 1978–1988, Association for Computing Machinery, 2023.

[43] "AWS Lambda." https://aws.amazon.com/lambda. Accessed Jan 2023.

[44] M. Malawski, A. Gajek, A. Zima, B. Balis, and K. Figiela, "Serverless execution of scientific workflows: Experiments with Hyperflow, AWS Lambda and Google Cloud Functions," *Future Generation Computer Systems*, vol. 110, pp. 502–514, 2020.

[45] R. Chard, Y. Babuji, Z. Li, T. Skluzacek, A. Woodard, B. Blaiszik, I. Foster, and K. Chard, "FuncX: A federated function serving fabric for science," in *29th International Symposium on High-performance Parallel and Distributed Computing*, pp. 65–76, 2020.

[46] A. Al-Saadi, D. H. Ahn, Y. Babuji, K. Chard, J. Corbett, M. Hategan, S. Herbein, S. Jha, D. Laney, A. Merzky, *et al.*, "Exaworks: Workflows for exascale," in *IEEE Workshop on Workflows in Support of Large-Scale Science*, pp. 50–57, IEEE, 2021.

[47] L. T. Meyer, M. Schouler, R. A. Caulk, A. Ribés, and B. Raffin, "High throughput training of deep surrogates from large ensemble runs," in *International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–16, 2023.

[48] M. Zvyagin, A. Brace, K. Hippe, Y. Deng, B. Zhang, C. O. Bohorquez, A. Clyde, B. Kale, D. Perez-Rivera, H. Ma, C. M. Mann, M. Irvin, J. G. Pauloski, L. Ward, V. Hayot, M. Emani, S. Foreman, Z. Xie, D. Lin, M. Shukla, W. Nie, J. Romero, C. Dallago, A. Vahdat, C. Xiao, T. Gibbs, I. Foster, J. J. Davis, M. E. Papka, T. Brettin, R. Stevens, A. Anandkumar, V. Vishwanath, and A. Ramanathan, "GenSLMs: Genome-scale language models reveal SARS-CoV-2 evolutionary dynamics," *The International Journal of High Performance Computing Applications*, vol. 37, no. 6, pp. 683–705, 2023.

[49] J. Guo, L. Ward, Y. Babuji, N. Hoyt, M. Williamson, I. Foster, N. Jackson, C. Benmore, and G. Sivaraman, "Composition-transferable machine learning potential for LiCl-KCl molten salts validated by high-energy X-ray diffraction," *Physical Review B*, vol. 106, no. 1, p. 014209, 2022.

[50] A. S. Villarreal, Y. Babuji, T. Uram, D. S. Katz, K. Chard, and K. Heitmann, "Extreme scale survey simulation with Python workflows," in *IEEE 17th International Conference on eScience*, pp. 206–214, IEEE, 2021.

[51] Y.-Y. Zhang, W. Gao, S. Chen, H. Xiang, and X.-G. Gong, "Inverse design of materials by multi-objective differential evolution," *Computational Materials Science*, vol. 98, pp. 51–55, 2015.

[52] Y. Dan, Y. Zhao, X. Li, S. Li, M. Hu, and J. Hu, "Generative adversarial networks (GAN) based efficient sampling of chemical composition space for inverse design of inorganic materials," *npj Computational Materials*, vol. 6, no. 84, 2020.

[53] T. Long, N. M. Fortunato, I. Opahle, Y. Zhang, I. Samathrakis, C. Shen, O. Gutfleisch, and H. Zhang, "Constrained crystals deep convolutional generative adversarial network for the inverse design of crystal structures," *npj Computational Materials*, vol. 7, no. 66, 2021.

[54] B. Kim, S. Lee, and J. Kim, "Inverse design of porous materials using artificial neural networks," *Science Advances*, vol. 6, no. 1, 2020.

[55] X. Huan, J. Jagalur, and Y. Marzouk, "Optimal experimental design: Formulations and computations," *Acta Numerica*, vol. 33, pp. 715–840, 2024.

[56] V. G. Satorras, E. Hoogeboom, and M. Welling, "E(n) equivariant graph neural networks," in *International Conference on Machine Learning*, pp. 9323–9332, 2021.

[57] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky, "E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials," *Nature communications*, vol. 13, no. 1, p. 2453, 2022.

[58] S. Axelrod and R. Gómez-Bombarelli, "GEOM, energy-annotated molecular conformations for property prediction and molecular generation," *Scientific Data*, vol. 9, no. 1, p. 185, 2022.

[59] N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch, and G. R. Hutchison, "Open Babel: An open chemical toolbox," *Journal of Cheminformatics*, vol. 3, p. 33, Dec. 2011.

[60] T. Halgren, "Merck molecular force field. II. MMFF94 van der Waals and electrostatic parameters for intermolecular interactions," *J Comput Chem*, vol. 17, pp. 520–552, 1996.

[61] G. Landrum, "Rdkit documentation," *Release*, vol. 1, no. 1-79, p. 4, 2013.

[62] M. O'Keeffe, M. A. Peskov, S. J. Ramsden, and O. M. Yaghi, "The Reticular Chemistry Structure Resource (RCSR) Database of, and Symbols for, Crystal Nets," *Accounts of Chemical Research*, vol. 41, pp. 1782–1789, Dec. 2008.

[63] A. Altomare, N. Corriero, C. Cuocci, A. Falcicchio, A. Moliterni, and R. Rizzi, "OChemDb: The free on-line Open Chemistry Database portal for searching and analysing crystal structure information," *Journal of Applied Crystallography*, vol. 51, pp. 1229–1236, 2018.

[64] R. Anderson, "cif2lammps." https://github.com/rytheranderson/cif2lammps.

[65] M. A. Addicoat, N. Vankova, I. F. Akter, and T. Heine, "Extension of the universal force field to metal–organic frameworks," *Journal of Chemical Theory and Computation*, vol. 10, no. 2, pp. 880–891, 2014.

[66] D. E. Coupry, M. A. Addicoat, and T. Heine, "Extension of the universal force field for metal–organic frameworks," *Journal of Chemical Theory and Computation*, vol. 12, no. 10, pp. 5215–5225, 2016.

[67] J. VandeVondele, M. Krack, F. Mohamed, M. Parrinello, T. Chassaing, and J. Hutter, "Quickstep: Fast and accurate density functional calculations using a mixed Gaussian and plane waves approach," *Computer Physics Communications*, vol. 167, no. 2, pp. 103–128, 2005.

[68] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical programming*, vol. 45, no. 1, pp. 503–528, 1989.

[69] J. P. Perdew, K. Burke, and M. Ernzerhof, "Generalized gradient approximation made simple," *Physical Review Letters*, vol. 77, p. 3865, May 1996.

[70] S. Goedecker, M. Teter, and J. Hutter, “Separable dual-space Gaussian pseudopotentials,” *Physical Review B*, vol. 54, pp. 1703–1710, Jul 1996.

[71] J. VandeVondele and J. Hutter, “Gaussian basis sets for accurate calculations on molecular systems in gas and condensed phases,” *The Journal of Chemical Physics*, vol. 127, p. 114105, Sept. 2007.

[72] S. Grimme, J. Antony, S. Ehrlich, and H. Krieg, “A consistent and accurateab initioparametrization of density functional dispersion correction (dft-d) for the 94 elements h-pu,” *The Journal of Chemical Physics*, vol. 132, Apr. 2010.

[73] T. A. Manz and N. G. Limasa, “Introducing DDEC6 atomic population analysis: Part 1. Charge partitioning theory and methodology,” *RSC Advances*, vol. 6, pp. 47771—47801, 2016.

[74] T. A. Manz and N. G. Limasa, “Introducing DDEC6 atomic population analysis: Part 2. Computed results for a wide range of periodic and nonperiodic materials,” *RSC Advances*, vol. 6, pp. 45727—45747, 2016.

[75] L. Ward, J. G. Pauloski, V. Hayot-Sasson, R. Chard, Y. Babuji, G. Sivaraman, S. Choudhury, K. Chard, R. Thakur, and I. Foster, “Cloud services enable efficient AI-guided simulation workflows across heterogeneous resources,” in *Heterogeneity in Computing Workshop*, New York, NY, USA: IEEE Computer Society, 2023.

[76] L. Ward, J. G. Pauloski, V. Hayot-Sasson, Y. Babuji, A. Brace, R. Chard, K. Chard, R. Thakur, and I. Foster, “Employing artificial intelligence to steer exascale workflows with Colmena,” *The International Journal of High Performance Computing Applications*, vol. 0, no. 0, p. 10943420241288242, 0.

[77] J. G. Pauloski, V. Hayot-Sasson, L. Ward, N. Hudson, C. Sabino, M. Baughman, K. Chard, and I. Foster, “Accelerating communications in federated applications with transparent object proxies,” in *International Conference for High Performance Computing, Networking, Storage and Analysis*, SC ’23, (New York, NY, USA), Association for Computing Machinery, 2023.

[78] J. G. Pauloski, V. Hayot-Sasson, L. Ward, A. Brace, A. Bauer, K. Chard, and I. Foster, “Object proxy patterns for accelerating distributed applications,” 2024. Preprint ArXiv:2407.01764.

[79] “NVIDIA Multi Process Service.” https://docs.nvidia.com/deploy/mps/index.html.

[80] E. C. R. Lopez and J. V. D. Perez, “CD-MOFs for CO2 capture and separation: Current research and future outlook,” *Engineering Proceedings*, vol. 56, no. 1, 2023.

[81] M. Witman, S. Ling, A. Gładysiak, K. Stylianou, B. Smit, B. Slater, and M. Haranczyk, “Rational design of a low-cost, high-performance metal-organic framework for hydrogen storage and carbon capture,” *The Journal of Physical Chemistry C*, vol. 121, 12 2016.

[82] J. Li, W. Ye, and C. Chen, “Chapter 5 - Removal of toxic/radioactive metal ions by metal-organic framework-based materials,” in *Emerging Natural and Tailored Nanomaterials for Radioactive Waste Treatment and Environmental Remediation* (C. Chen, ed.), vol. 29 of *Interface Science and Technology*, pp. 217–279, Elsevier, 2019.

[83] M. Safaei, M. M. Foroughi, N. Ebrahimpoor, S. Jahani, A. Omidi, and M. Khatami, “A review on metal-organic frameworks: Synthesis and applications,” *TrAC Trends in Analytical Chemistry*, vol. 118, pp. 401–425, 2019.

[84] D. Li, A. Yadav, H. Zhou, K. Roy, P. Thanasekaran, and C. Lee, “Advances and applications of metal-organic frameworks (MOFs) in emerging technologies: A comprehensive review,” *Global Challenges*, vol. 8, no. 2, p. 2300244, 2024.

[85] L. Wang, Z. Zhou, X. Yang, S. Shi, X. Zeng, and D. Cao, “The present state and challenges of active learning in drug discovery,” *Drug Discovery Today*, p. 103985, 2024.