Nanoporous Materials Discovery via Search Bias-Guided Surrogate Modeling

Azza Fadhel[1], Yassine Chemingui[1], Minh Hoang[2], Aryan Deshwal[3], Trong Nghia Hoang[1], Jana Doppa[1]

[1]Washington State University,
[2]Princeton University,
[3]University of Minnesota, Twin Cities

{azza.fadhel, yassine.chemingui, trongnghia.hoang, jana.doppa}@wsu.edu, minhhoang@princeton.edu, adeshwal@umn.edu

Abstract

Nanoporous materials (NPMs) are suitable for solving some of the society's biggest challenges including carbon capture and conversion, storing hydrogen and methane, and sensing gases. The key challenge in discovering high-performing NPMs for a target application is that making and evaluating candidate NPMs requires performing resource-expensive wet-lab experiments. We consider the problem of discovering NPMs using existing experimental data of NPM evaluations. The overall goal is to find better NPMs than the best NPMs from the past experimental data. A simple approach is to create a surrogate model to match the objective values on the given dataset and employ it to score candidate NPMs to discover optimized NPMs. However, this surrogate model will fail because it does not have the appropriate search bias for the goal of optimization. To address this challenge, we propose a novel surrogate modeling approach that combines value matching loss with an optimization bias regularizer. The key idea is to algorithmically realize search bias is to mimic the search behavior of monotonically increasing sequences of NPMs from the given dataset. Experiments on multiple real-world NPM discovery tasks demonstrate that our proposed surrogate model discovers significantly better NPMs than baselines including value matching surrogate model and one-step Bayesian optimization.

Code — https://github.com/azzafadhel/OptReg

Introduction

Nanoporous materials (NPMs) (Yaghi 2019), which are three-dimensional crystals, have the highest surface areas known to date. Due to the selective gas adsorption properties of NPMs, they enable a huge number of real-world applications in the storage, separation, and sensing of gases. Examples to demonstrate their potential to solve some of the society's biggest challenges include capturing carbon dioxide from air (Sumida et al. 2012) and sequestering it to prevent global warming (Trickett et al. 2017); storing hydrogen gas for fuel for hydrogen-powered vehicles to enable clean energy (Suh et al. 2012); and enabling electronic noses to detect toxic compounds and explosives (Yuan et al. 2022). Several families of NPMs (Furukawa et al. 2013; Feng, Ding, and Jiang 2012; Yang et al. 2023) are synthesized modularly by stitching together molecular building blocks. By using multiple topologies and molecular binding blocks, we can modularly synthesize a huge number of candidate NPMs exhibiting varying adsorption properties as shown in Fig 1.

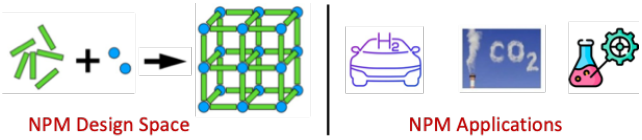

Figure 1: NPMs design space and their applications in hydrogen-powered cars, carbon capture, and drug delivery.

For a given real-world application, our goal is to find NPMs with maximum (adsorption) property value from a large set of candidate NPM structures (aka NPM discovery task). The key challenge in solving this NPM discovery task is that synthesizing and measuring the target property of a candidate NPM is resource-expensive, both in terms of human labor and raw physical materials. One popular approach to address this challenge is through the use of Bayesian optimization (BO) (Garnett 2023; Eriksson et al. 2019; Deshwal et al. 2023; Deshwal and Doppa 2021; Deshwal, Belakaria, and Doppa 2021b,a; Deshwal et al. 2022) which relies on the principle of active learning. The key idea behind BO is to build a statistical model from the past NPM evaluations and use it to intelligently select the sequence of candidate NPMs for evaluation in an iterative manner to quickly find NPMs with high property value by minimizing the resource cost of NPM evaluation experiments (Deshwal, Simon, and Doppa 2021; Gantzler et al. 2023).

In many real-world scenarios, while we do have access to an existing database of training examples in the form of NPM structures and property evaluations, performing online experiments following the iterative recommendations of BO will not be feasible due to the prohibitively expensive cost of materials and equipment to perform new NPM evaluations. To avoid repeating such experimental overhead across a large number of sequential online experiments, we instead investigate an alternative (but more affordable) approach which focuses on building a surrogate model that can rank well the performance of unevaluated NPM structures from a given candidate set. One naive approach is to create

a surrogate model (e.g., neural network) by optimizing the mean squared error (MSE) loss (Duda, Hart et al. 2006) on the past experimental data and use its prediction to score the candidate NPMs in a given set (Chong et al. 2020; Moosavi, Jablonka, and Smit 2020; Mukherjee and Colón 2021). A batch of high-scoring candidate NPMs can then be selected.

However, this approach is driven by a prediction bias that focuses on minimizing the average prediction error on unknown NPMs, which is not as important as learning a scoring function that preserves their actual (unobserved) performance ranking, especially with limited training data. In fact, the optimization task needs a scoring function that preserves such performance ranking rather than the one that provides an exact prediction. This raises the following question:

How can we mitigate the over-specialization to such prediction bias while fitting a surrogate model to training data?

To address this question, we develop a novel surrogate model training approach that incorporates an optimization bias as a regularizer to the training loss (MSE). Specifically, the goal of optimization regularizer is to enable the surrogate model to rank unevaluated NPMs as per their true property values. Our approach is related to the general area of learning to rank and preference learning (Joachims 2002; Herbrich, Graepel, and Obermayer 2000; Burges et al. 2005; Burges 2010; Doppa, Fern, and Tadepalli 2014a,b). However, our formulation of the regularizer encourages matching the gradient of the surrogate and the true property function using a training dataset of NPM and true property value pairs. To practically achieve this goal by implementing the optimization bias, we propose an algorithmic approach to mimic the search behavior of monotonically increasing sequences of NPMs from the given training dataset (experimental evaluations of a set of NPMs).

Our experiments on multiple real-world NPM discovery tasks demonstrate that the proposed surrogate modeling approach outperforms two representative prediction-based surrogate baselines. We also analyze the different features of NPMs to gain insights about which features are critical for high-property values. We found that the results from this analysis were different for diverse NPM discovery tasks. These insights can be useful for domain scientists in improving their understanding of structure-property relationships in NPMs.

Problem Setup and Challenges

Suppose $\mathcal{X}$ is an input space of NPM structures where each $x \in \mathcal{X}$ is a candidate NPM. Without loss of generality, we assume that each candidate NPM x can be represented as a d-dimensional feature vector. This representation can be created manually by materials science experts or automatically by deep generative models from a given database of unsupervised NPM structures (i.e., no NPM property values).

Let $f : \mathcal{X} \mapsto \Re$ be an unknown, expensive real-valued objective function which can evaluate any given NPM structure $x \in \mathcal{X}$ to produce output $y = f(x)$. For NPM property evaluation, $f(x)$ corresponds to running a resource-expensive physical lab experiment which includes synthesizing the NPM and evaluating its property. The resource expense of this experiment includes human labor and cost of raw materials. Our goal is to find an NPM structure $x \in \mathcal{X}$ that approximately optimizes the true property f:

$$\hat{x} \quad \text{s.t.} \quad f(\hat{x}) \approx \max_{x} \; f(x) \tag{1}$$

We consider the following variant of this optimization problem. We are provided with a database of n NPM structure and property evaluation pairs $\mathcal{D}$=$\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$ collected from past experiments, where y_i=$f(x_i)$. We do not have access to objective function f values on NPMs outside the dataset $\mathcal{D}$. We are given a set of m candidate NPM structures without property evaluations $\mathcal{C}$=$\{x_1, x_2, \cdots, x_m\}$ (i.e., unevaluated NPMs). The overall goal of the NPM discovery task is to solve the above optimization problem over the candidate set $\mathcal{C}$. An algorithm to solve NPM discovery task produces an input $\hat{x} \in \mathcal{C}$. We measure the accuracy of solution in terms of the real property value of $\hat{x}$, namely $f(\hat{x})$. Ideally, $f(\hat{x})$ should be higher than the best property value seen in the training dataset $\mathcal{D}$ (say $\mathcal{D}(\mathbf{best})$=$\max\{y_1, y_2, \cdots, y_n\}$).

From a real-world deployment perspective, the workflow is as follows. The algorithm selects b NPMs from the candidate set $\mathcal{C}$ for experimental evaluation in a batch. It is possible that some selected NPMs are not synthesizable. Therefore, the experimental material scientists would prefer to have several NPMs with high property values in this set of b NPMs. For these reasons, we employ both *maximum* and *median* statistics over the objective values of the selected NPMs to measure the efficacy of NPM discovery algorithms.

Challenge. The key challenge in solving NPM discovery tasks is to train a surrogate model that can reliably score the unevaluated NPMs in the candidate set $\mathcal{C}$ that aligns with the goal of optimization (i.e., finding high-performing NPMs).

There are two ways to approach materials discovery problems: 1) Searching a large fixed set of libraries, and 2) Generating de-novo from deep generative models DGMs which is still in its infancy. (Szymanski and Bartel 2025) showed that even simple template matching outperforms DGMs. In both these approaches, the key challenge is expensive experimentation: selecting a small pool most likely to contain the materials with high property values. Our surrogate modeling approach precisely addresses this challenge.

Technical Approach

In this section, we first provide the details of three representative baseline approaches to solve the nanoporous materials discovery problem. Next, we describe our proposed surrogate modeling approach that overcomes the drawbacks of these baselines by incorporating appropriate optimization bias as a regularizer during training. See Fig. 2 for an overview of our surrogate-guided NPM discovery approach.

Surrogate Baselines

Surrogate model guided discovery methods can be specified with two key components: *1. Train a surrogate model* on the existing dataset of evaluated NPMs $\mathcal{D}$. *2. Score and rank candidates* by using the trained model to compute a score for

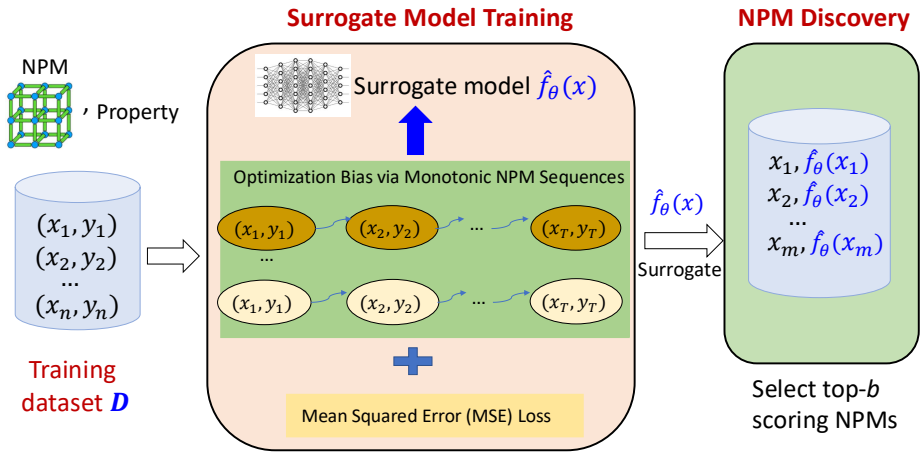

Figure 2: High-level overview of the proposed surrogate model-guided NPM discovery framework.

each unevaluated candidate in the candidate set $\mathcal{C}$ and selecting the top-ranked ones. We employ two baseline methods which instantiate this recipe as follows:

1. **One-step Bayesian Optimization (BO):** This approach uses a Gaussian Process $\hat{f}$ model as the surrogate model. The score for each candidate NPM $x \in \mathcal{C}$ is given by an acquisition function (concretely Expected Improvement $EI(\hat{f}, x)$), which balances the predicted property value (exploitation) with the model's uncertainty (exploration).
2. **Naive Surrogate Model (MSE Loss):** This baseline trains a neural network $\hat{f}(x, \theta)$ as the surrogate model by formulating it as a regression learning problem and minimizing the MSE loss function: $\mathcal{L}_{\text{MSE}}(\theta) = \sum_{(x_i, y_i) \in \mathcal{D}} \left(\hat{f}(x_i, \theta) - y_i\right)^2$.
   Here, the score for each candidate NPM $x \in \mathcal{C}$ is simply the model's prediction of the property value $\hat{y}$=$\hat{f}(x, \theta)$.

Batch BO Baseline. We also consider batch BO acquisition strategy (e.g., q-EI) as a baseline. However, the key challenge is acquisition function optimization since we need to perform combinatorial search for Top-b NPMs (for a batch size of b) over a discrete set of n NPMs. We employ randomized local search to solve it.

Proposed Surrogate Modeling Approach

To solve the NPM discovery task effectively, the mechanism to score candidate NPMs using the surrogate model should preserve the ranking order from the true objective function f (i.e., true property values). Unfortunately, the above-mentioned naive surrogate model approach does not align well with this goal as it was designed to predict the true property values by minimizing the MSE loss which is a much harder and unnecessary problem given the limited size of the training dataset $\mathcal{D}$. Although the one-step and batch BO approaches would be slightly better since utility functions in BO directly target optimization but they can be too explorative and can suffer in some NPM discovery tasks because BO works under the assumption of iterative training data to update its beliefs via surrogate model. Indeed, our experiments on NPM discovery tasks demonstrate these drawbacks of the baseline methods.

To overcome the drawbacks of these three baseline methods, we propose a novel surrogate modeling approach that instead aims to learn the gradient field of the oracle function rather than predict its output. The key insight here is that any function that share the same gradient field with the oracle will also preserve its ranking order on the NPM design space. For example, functions that differ from the oracle by a constant share the same maxima despite having different maximum values. This means learning to predict its output is an unnecessary design bias. Furthermore, learning the oracle's gradient field captures the entire geometric structure of improvement directions, providing a globally consistent and differentiable notion of preference. In contrast, ranking-based methods must approximate the non-differentiable global ranking loss with local (pairwise or listwise) surrogate losses, introducing additional modeling gaps that might weaken generalization, especially when training data is limited. A thorough comparison with ranking-based approach is however orthogonal to the main scope of our work and is deferred to future work.

To substantiate the above insight, we will (1) sample sequences of NPMs from the offline dataset $\mathcal{D}$ with increasing property values; and (2) fit a surrogate's gradient field to these sequences via minimizing how the gradient flow from one NPM design deviates from its next design in a sampled sequence. Below we describe the details of these two key steps of our approach.

Creating Monotonic NPM Sequences. We create several monotonically increasing sequences of NPMs with length T as follows. We first create T bins of the NPMs in the training data $\mathcal{D}$ based on their percentiles. Next, we sample one NPM

Algorithm 1: Surrogate Training with Optimization Bias

Require: NPM training data $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, candidate set $\mathcal{C} = \{x_1, x_2, \ldots, x_m\}$, hyperparameter λ, number of monotonic sequences T

Ensure: Top-b NPMs based on surrogate prediction

```
Initialize surrogate model f̂(x, θ) parameters θ
Create monotonic NPM sequences from D: S = {S_1, S_2, ..., S_T}
repeat
    for each monotonic sequence S_i ∈ S do
        for each consecutive pair (x, x') ∈ S_i do
            Compute regularizer L_OPT(θ) using Eq. (4)
        end for
    end for
    Compute MSE loss L_MSE(θ) on D using Eq. (2)
    Combine losses: L(θ) = L_OPT(θ) + λ · L_MSE(θ)
    Update surrogate parameters θ to minimize L(θ)
until convergence or maximum iterations
Use surrogate model f̂(x, θ) to score each x ∈ C
Select top-b highest scoring NPMs from C
return top-b NPM candidates from C
```

from each bin to create a sequence of NPMs of length T with increasing adsorption property values. We repeat this sampling procedure multiple times to create several monotonic NPM sequences $\mathcal{S}$, which are intended to provide training data for the behavior of local search based optimization.

Optimization Regularizer to Mimic Search Behavior. Technically, this step corresponds to approximating the gradient field of the true property function f using the monotonic NPM sequences. This is a challenging problem because we are only given the property values for a fixed set of NPMs (i.e., training dataset $\mathcal{D}$) and we do not have the ability to query f on additional NPMs to apply finite difference methods (Iserles 2009) for approximating its gradient.

To overcome this challenge, we will instead make use of the scalar difference $\Delta f = f(x) - f(x')$ and note that for an interpolant path $x(t)$ connecting x and x', we have

$$\Delta f \;=\; \int_0^1 \frac{\mathrm{d}}{\mathrm{d}t} f(x(t))\mathrm{d}t = \int_0^1 \nabla f(x(t))^\top \frac{\mathrm{d}}{\mathrm{d}t} x(t)\mathrm{d}t\,. \quad (2)$$

As this is true regardless of how we choose the interpolant paths $x(t)$, it reveals a novel learning paradigm where we can recover the unknown target property function's gradient field via designing an appropriate interpolant path that connects the observed offline data points, while encoding prior knowledge within its local geometries $dx(t)/dt$. This perspective naturally aligns with an amortized learning principle where the surrogate gradient field is shaped by aggregated constraints from many fragments of local geometries. Even if some local geometries along the interpolant path are inaccurate, the correct or informative fragments will intuitively occur more frequently across the interpolant path. As a result, it creates a training signal that implicitly prioritizes incorporating relevant fragments to recover the true gradient field, while marginalizing out irrelevant ones. In the scope of this paper, we choose the linear interpolation design $x(t) = (1-t)x + tx'$ to instruct the learning algorithm on recovering all fragments of linear geometries around the observed data. Despite the simplicity, we note that most sophisticated geometric structures can be decomposed into fragments of (possibly infinitesimal) linear geometries. Thus, this suffices to recover the true gradient field assuming it is sufficiently smooth. Plugging this design into Eq. (2), we obtain

$$\begin{aligned}\Delta f \;&=\; f(x) - f(x') \;=\; \Big(x' - x\Big)^\top \int_0^1 \Big[\nabla f\Big(x(t)\Big)\Big]\mathrm{d}t \\ &\simeq \Big(x' - x\Big)^\top \int_0^1 \Big[\nabla f_\theta\Big(x(t)\Big)\Big]\mathrm{d}t\,, \end{aligned} \quad (3)$$

where $x(t) = x \cdot (1-t) + x' \cdot t$. Therefore, to incorporate optimization/search bias using the monotonic NPM sequences $\mathcal{S}$, we need to find parameters θ by minimizing the following loss function over *every consecutive pair of NPMs* (x, x') from each monotonic NPM sequence $S_i \in \mathcal{S}$.

$$L_{\mathrm{OPT}}(\theta) = \frac{1}{2}\mathbb{E}\left(\Delta f - (x'-x)^\top \int_0^1 \Big[\nabla f_\theta\Big(x(t)\Big)\Big]\mathrm{d}t\right)^2$$

where the expectation is approximated via summation over a discrete set of samples. For more insight, we prove that minimizing Eq. (3) is equivalent to minimizing the variance on the difference between the (unknown) target function and surrogate which in fact encourages their gradient field to overlap as stated below.

Theorem 1 *Suppose the surrogate is parameterized with θ as stated in Eq.* (3)*, we have $L_{\mathrm{OPT}}(\theta) = V[f(x) - f_\theta(x)]$.*

Proof: Recall that we can rewrite

$$L_{\mathrm{OPT}}(\theta) = \frac{1}{2}\mathbb{E}\left(\Delta f - (x'-x)^\top \int_0^1 \Big[\nabla f_\theta\Big(x(t)\Big)\Big]\mathrm{d}t\right)^2 \quad (4)$$

Next, we note that $x(t) = x \cdot (1-t) + x' \cdot t$ and hence, by the line integration theorem,

$$f_\theta(x) - f_\theta(x') = (x'-x)^\top \int_0^1 \Big[\nabla f_\theta\Big(x(t)\Big)\Big]\mathrm{d}t\,. \quad (5)$$

Plugging the above and the fact that $\Delta f = f(x) - f(x')$ into Eq. (4), we can rewrite it as

$$L_{\mathrm{OPT}}(\theta) = \frac{1}{2}\mathbb{E}\left(\big(f(x) - f_\theta(x)\big) - \big(f(x') - f_\theta(x')\big)\right)^2 \quad (6)$$

Re-arranging Eq. (6) and noting that x and x' are statistically identical, we have

$$\begin{aligned} L_{\mathrm{OPT}}(\theta) \;&=\; \mathbb{E}[(f(x) - f_\theta(x))^2] - \mathbb{E}^2[(f(x) - f_\theta(x))] \\ &=\; \mathbb{V}[(f(x) - f_\theta(x))]\,, \end{aligned} \quad (7)$$

where the last step follows from the definition of variance. This completes our proof.

$$L(\theta) \;=\; L_{\mathrm{OPT}}(\theta) \;+\; \lambda \cdot L_{\mathrm{MSE}}(\theta) \quad (8)$$

The training objective to create surrogate model is a combination of the MSE loss and the optimization regularizer (OptReg) loss. The hyper-parameter λ allows us to trade-off matching property values and matching optimization bias. More intuitively, we can think of the MSE regularization as an implicit mechanism to encode prior knowledge in the local geometries of the interpolant design $x(t)$. To summarize, our proposed approach has three key steps.

- Train surrogate $\hat{f}(x, \theta)$ by minimizing loss L in Eq. (8).
- Score each NPM $x \in \mathcal{C}$ using the prediction $\hat{y}$=$\hat{f}(x, \theta)$.
- Select the top-b highest scoring NPMs as the output.

Experiments and Results

This section describes our experimental evaluation of the proposed surrogate modeling approach and the five baseline methods on multiple real-world NPM discovery tasks.

Experimental Setup

NPM Discovery Tasks. We consider three diverse real-world NPM discovery tasks for our experimental evaluation.

I. Hydrogen Storage Tasks: The first two tasks focus on predicting hydrogen storage capacities in metal-organic frameworks (MOFs), an important class of NPMs as described in (Ahmed and Siegel 2021). The two tasks involve predicting two key properties of MOFs: usable gravimetric capacity (UG) and usable volumetric capacity (UV). We refer to the two tasks as **MOF (UG)** and **MOF (UV)**. These properties are critical for assessing a MOF's potential for hydrogen storage applications, particularly in the context of fuel cell vehicles. The expert-designed MOF representation for these two tasks consists of seven features: single-crystal density (d), pore volume (pv), gravimetric surface area (gsa), volumetric surface area (vsa), void fraction (vf), largest cavity diameter (lcd), and pore limiting diameter (pld).

II. Natural Gas Storage Task: The third task involves predicting the methane deliverable capacity of Covalent Organic Frameworks (COFs), another important class of NPMs as described in (Mercado et al. 2018). This task, referred as **COF(NG)**, is critical for identifying optimal materials for vehicular adsorbed natural gas storage.

The expert-designed COF representation consists of 12 structural and chemical features which include void fraction, density, largest included sphere diameter, largest free sphere diameter, gravimetric surface area, and densities of various elements (carbon, fluorine, hydrogen, nitrogen, oxygen, sulfur, and silicon). The target property, methane deliverable capacity, is measured in L STP CH_4/L COF at 298 K under a 65 bar to 5.8 bar pressure swing. This property is crucial as it primarily determines the driving range of a vehicle using an adsorbed natural gas fuel tank packed with the COF.

Configuration of Methods. We configure the baselines and our proposed approach as follows.

Naive (MSE) Surrogate Model: We employ a standard neural network as a surrogate model, trained with MSE loss. We keep the architecture and other hyper-parameters for this baseline exactly the same as our proposed method.

One-step Bayesian Optimization: We implement the one-step BO method as described in technical section using the popular BoTorch library (Balandat et al. 2020). We employ Gaussian process surrogate model with radial basis function kernel and expected improvement as the utility function.

Batch BO (q-EI): We employ q-EI as the batch acquisition strategy and perform randomized local search for 15 iterations to select a batch of b NPMs.

Deep Generative Model: We consider a generative modeling based solution where we fit a denoising diffusion model (Ho, Jain, and Abbeel 2020) on monotonically increasing NPM sequences and select the candidate NPMs by conditioning on the best objective value from the given training set.

Evolutionary Search: CMA-ES is a strong gradient-free optimization baseline (Hansen and Ostermeier 2001). It samples candidates from a multivariate normal distribution and iteratively adapts its mean, covariance matrix, and step size toward regions associated with higher property values.

Proposed Surrogate Model (MSE + OptReg): We employ a standard neural network and train it to optimize the combined MSE loss with optimization bias regularizer using Algorithm 1. We used a pyramid structured feed-forward neural network with 4 layers ($512 \rightarrow 128 \rightarrow 32 \rightarrow 1$). For each task, we trained the model using Adam optimizer (1e-4 learning rate) and a batch size of 128 for 100 epochs. We found the experiments to be robust to choice of λ (regularization parameter) values and set λ=1 for all final results.

Evaluation Methodology. To simulate real-world experimental conditions, we utilize a large dataset of NPM and adsorption property pairs, denoted as $\mathcal{D}_{full}$, for each NPM discovery task. We first rank this dataset in ascending order of property values. The *lowest-ranked* subset of size p is then selected to represent the existing experimental dataset $\mathcal{D}$, which we refer to as the training data. This mimics the scenario where initial experiments have been conducted on a limited set of materials.

We evaluate our proposed method and the baselines by using them to rank the NPM structures in the remaining dataset, denoted as $\mathcal{C}$=$\mathcal{D}_{full} \setminus \mathcal{D}$. From these rankings, we select the top-b ranked NPMs. This approach closely resembles the real-world scenario where materials researchers would experimentally evaluate a set of materials from those that are experimentally feasible, based on predictions or rankings from their models. This methodology allows us to assess the effectiveness of different approaches in identifying high-performing NPMs while minimizing the number of experimental evaluations required. We denote $\mathcal{D}$(**best**) as the best NPM from the training data $\mathcal{D}$. The effectiveness of any given approach can be measured by how better the discovered NPMs are compared to $\mathcal{D}$(**best**).

Results and Discussion

Comparison with Baselines. We compare our proposed method OptReg with the baselines by examining the property values in the top b ranked NPMs for each method, using a training dataset size of $n = 3000$ samples. Table 1 (and 4 in appendix) illustrate this for b=100 for the three tasks COF,

Method	COF(NG)	MOF(UG)	MOF(UV)
$\mathcal{D}$(best)	87.748	0.7	12.1
MSE Surrogate	117.906 ± 2.084	$\mathbf{7085.714} \pm 0.00$	204.545 ± 49.317
One-Step BO	15.698 ± 0.00	297.003 ± 0.00	198.389 ± 0.00
Batch BO (q-EI)	108.717 ± 2.465	6887.462 ± 0.00	182.089 ± 0.00
CMA-ES	115.153 ± 7.186	2432.618 ± 62.942	292.066 ± 11.980
Diffusion Model	65.856 ± 22.549	204.853 ± 10.976	174.740 ± 12.800
OptReg (Ours)	$\mathbf{125.805} \pm 0.00$	$\mathbf{7085.714} \pm 0.00$	$\mathbf{309.091} \pm 0.00$

Table 1: Optimization performance (max) across NPM discovery benchmarks using a dataset size of n=3000 and evaluation batch size b=100. Values denote the maximum improvement in property value over the best training material (mean ± std over 4 runs).

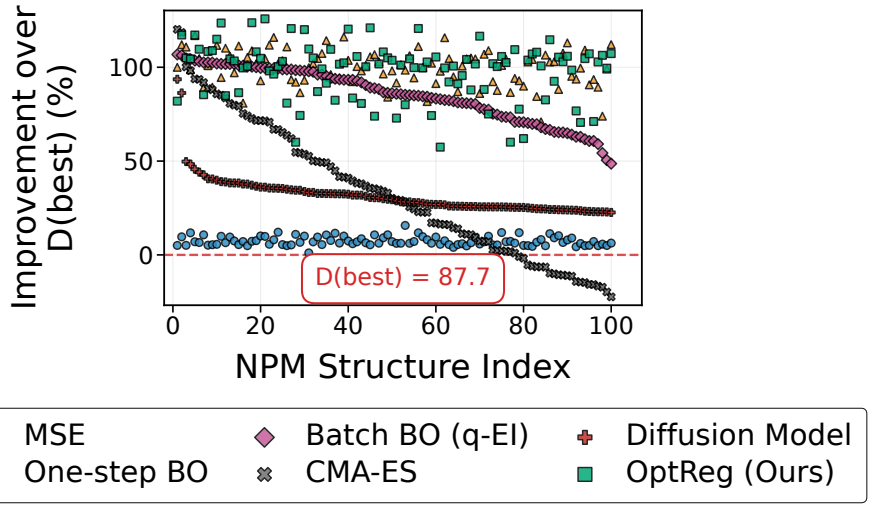

Figure 3: Performance comparison of optimization methods on the COF(NG) discovery task with $n = 3000$ offline samples and $b = 100$ oracle evaluations.

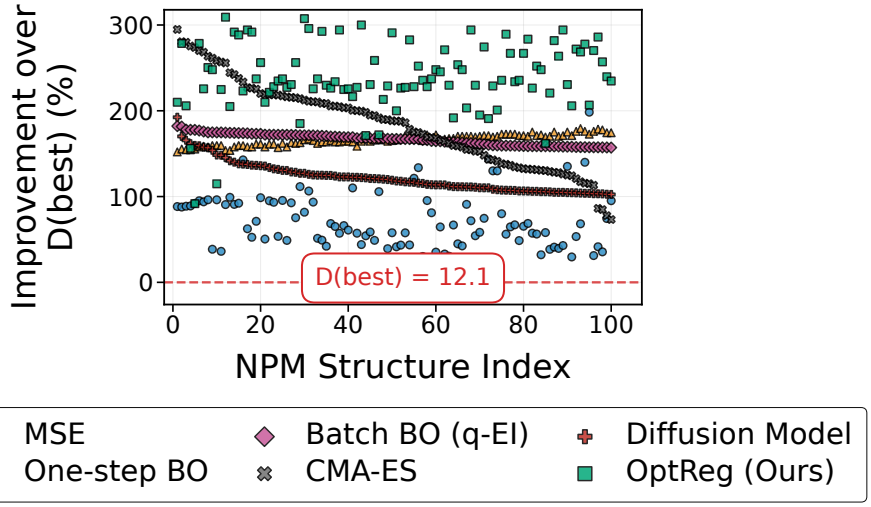

Figure 4: Performance comparison of optimization methods on the MOF(UV) discovery task with $n = 3000$ offline samples and $b = 100$ oracle evaluations.

MOF-UG, and MOF-UV. Each cell in the method columns list the percentage improvement over $\mathcal{D}_{train}$**(best)**, which

refers to the best absolute objective value in the given training dataset $\mathcal{D}$. Our method consistently outperforms both the baselines on the maximum and median value in the top b selected NPMs and discovers candidate NPMs with property values much higher than in the given training set.

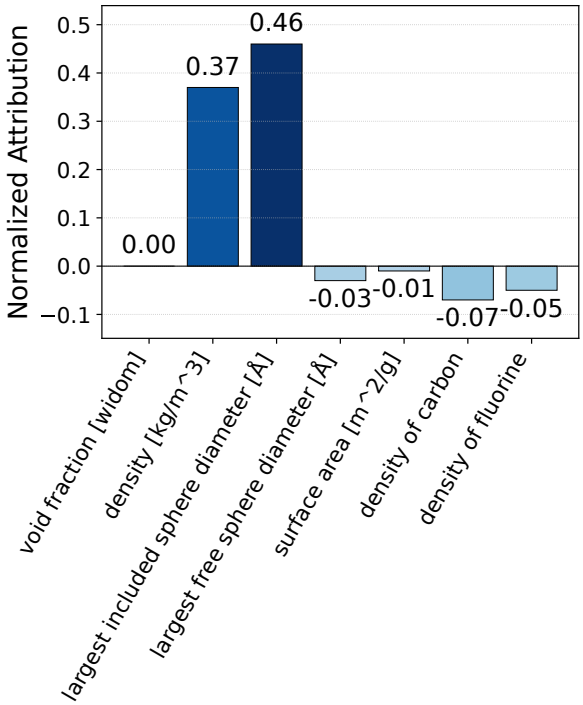

Figure 5: Feature importance analysis for MOF(UG) task.

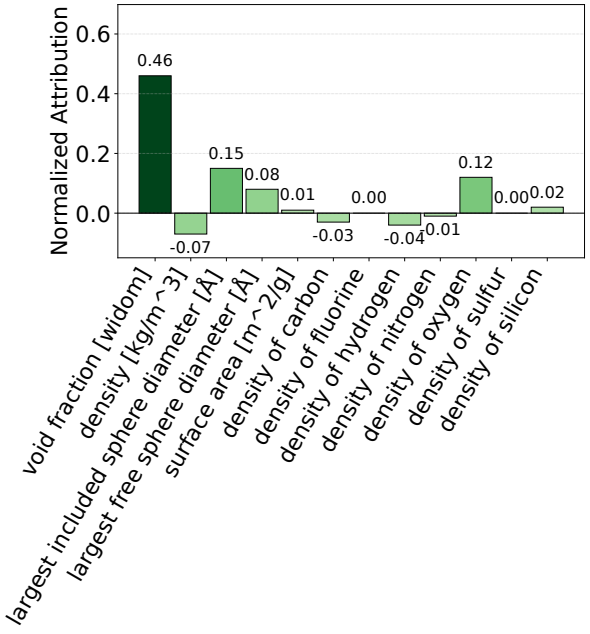

Figure 6: Feature importance analysis for COF task.

The significance of these results are two fold. First, the discovered NPMs enabled much more cost-effective clean energy solutions (e.g., hydrogen-powered vehicles). Second, better median values mean that we have more back up options if some top NPMs turn out to be not synthesizable. These results highlight the effectiveness of our method for NPM discovery tasks. The consistent large margin between our method and MSE baseline demonstrates that the optimization bias regularizer is effective and more robust.

Figures 3, 4 and (8 in appendix for space restriction) plots all the top $b = 100$ ranked NPMs for each method showing the percentage improvement in property values over $\mathcal{D}_{train}$**(best)** achieved by each NPM. We prefer the set of selected NPMs to be in regions of higher property values since some materials might fail on unseen but important objectives (e.g., synthesizability) not included during the optimization. Our method consistently results in better or similar distribution compared to the five baselines demonstrating its practical applicability.

Feature Attribution Analysis. Figures 5, 6 and 7, shows the feature importance scores for the three NPM discovery tasks based on Integrated Gradients (IG) method (Sundararajan, Taly, and Yan 2017). The IG method provides a way to attribute a deep network's prediction to its input features by accumulating gradients along a straight-line path from a baseline input to the actual input. The approach is built upon two important axiomatic properties: Implementation Invariance (attributions are identical for functionally equivalent models) and Sensitivity (features that contribute differently to the output receive different attributions).

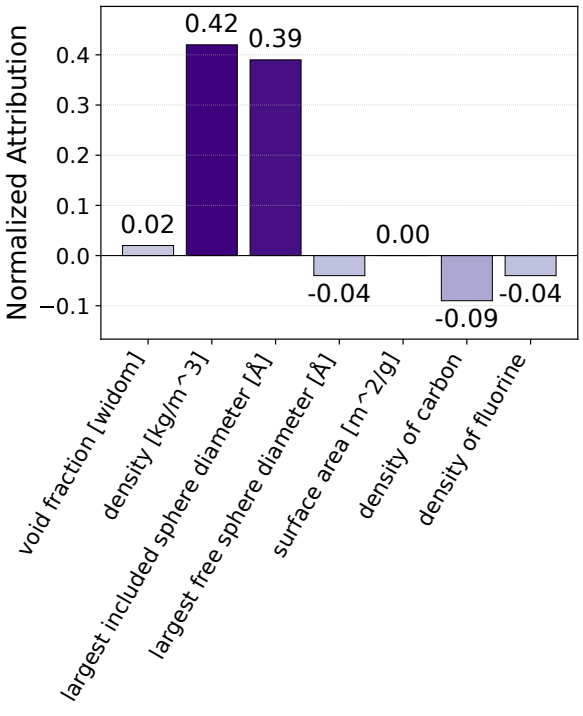

Figure 7: Feature importance analysis for MOF(UV) task.

Surrogate	COF	MOF_ug	MOF_uv
Linear	124.27 ± 0.16	4489.29 ± 31.75	257.23 ± 0.80
Non-Linear	$\mathbf{125.81} \pm 0.00$	$\mathbf{7085.71} \pm 0.00$	$\mathbf{309.10} \pm 0.00$

Table 2: Effect of replacing the neural surrogate with a linear model for training dataset size $n = 3000$ and evaluation batch size $b = 100$. Higher is better.

NPM Task	λ=0.1	λ=1	λ=10
COF	125.80 ± 0.00	125.80 ± 0.00	121.03 ± 0.00
MOF_uv	301.86 ± 0.27	309.09 ± 0.00	309.09 ± 0.00

Table 3: Effect of varying the regularization parameter λ for $n = 3000$ and $b = 100$ (higher is better).

We found that the *largest included sphere diameter* showed the highest attribution for property prediction in MOF(UG) task (0.46), indicating pore size as the primary performance driver. Material *density* had the second highest attribution (0.37), while *void-fraction* showed minimal impact. Similarly, both density and largest included sphere diameter were also the top ranked factors in MOF(UV) task, with attributions of 0.42 and 0.39 respectively. However, void fraction showed highest attribution score of 0.46 for the COF (NG), followed by the largest included sphere diameter (0.15) feature. Although these results demonstrate distinct structural features are informative in predicting properties for the two material systems (MOFs and COFs), all of these features are related to the pore architecture and overall material packing, both of which drive the adsorption property in this class of materials. This analysis is insightful from the perspective of a materials scientist as it improves their understanding of the structure-property relationships.

Ablation Studies. We conduct ablation experiments on two key parameters: varying the training dataset size n and the number of top-ranked NPMs b. Tables 6 and 13 (Tables shown in Appendix due to space constraints) shows the performance of our method and baselines as a function of n and b. Our approach consistently outperforms baselines across different values of n. This suggests that our method leverages the training dataset more effectively. Varying b reveals that our method maintains its superiority even when considering a larger set of top-ranked NPM candidates, indicating robustness in its rankings, especially, for MOF tasks.

We also perform an ablation on the complexity of surrogate model by comparing neural surrogate with a linear surrogate. Results in Table 2 demonstrate that OptReg with a linear surrogate performs substantially worse (vs. OptReg with a non-linear surrogate) and exhibits higher variance, demonstrating that nonlinear surrogates are necessary to capture feature-property relationships in NPM discovery.

Effect of the Regularization Parameter λ. We study the influence of the optimization-bias regularization parameter λ. Since λ is applied to the value-matching (MSE) component, larger values of λ cause the surrogate to prioritize exact value prediction, which reduces optimization performance. As shown in Table 3, OptReg remains stable for $\lambda \in [0.1, 10]$.

Roadmap to Deployment

The main goal of this work was to build the foundation for a principled and effective surrogate model training framework for NPM discovery tasks. Our ultimate aim is to develop a robust and efficient AI system for identifying high-performing NPMs for real-world applications with high societal impact, including carbon capture and conversion, hydrogen and methane storage, and gas sensing.

Collaboration with domain experts and key trade-off: The current study focused on employing real-world datasets of NPM experimental evaluations to develop and evaluate the effectiveness of surrogate model-guided NPM discovery methods before deploying in real-world scenarios. Based on positive results in this paper and discussions with our material science collaborators on our structure-property relationship analysis, we are currently working towards deploying our approach in a physical laboratory at Indiana University (Figure 2). A good feature of our data-driven approach is that it can be naturally integrated with experimental workflows: the algorithm produces b candidate NPMs for experimental evaluation and the domain scientists conduct physical lab experiments in batch to evaluate the selected NPMs. According to our domain collaborator, batch experimentation is a key requirement in NPM discovery since it typically involves significant advance planning and resource allocation.

After a pilot study for NPM discovery, we will extend the deployment for the discovery of water-stable Metal-Organic Polyhedra for aqueous separation applications such as the purification and treatment of urban wastewater, the remediation of contaminated groundwater, rivers, lakes, and soils.

Conclusions and Future Work

Nanoporous materials (NPMs) are enablers for solving sustainability challenges including carbon capture, clean energy, and drug delivery. However, discovering high-performing NPMs for a target application is challenging due to the need to perform expensive lab experiments to evaluate NPMs. We studied a principled surrogate modeling approach from past NPM experimental evaluation data towards the goal of selecting candidate NPMs with high performance outside the training data. Our results on three real-world tasks demonstrated high-efficacy of our surrogate model-guided NPM discovery approach. Our fine-grained analysis contributed to improved understanding of the structure-property relationships of NPMs.

Future work should integrate our approach into the experimental workflows of physical laboratories. Another promising research direction is to generalize the proposed surrogate for multi-objective optimization of nanoporous materials (e.g., CO_2 adsorption capacity, stability, and synthesis cost for gas separation applications) and comparing it with Bayesian optimization based approaches (Belakaria, Deshwal, and Doppa 2019; Belakaria et al. 2020).

References

Ahmed, A.; and Siegel, D. J. 2021. Predicting hydrogen storage in MOFs via machine learning. *Patterns*, 2(7).

Balandat, M.; Karrer, B.; Jiang, D.; Daulton, S.; Letham, B.; Wilson, A. G.; and Bakshy, E. 2020. BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. *Advances in neural information processing systems*, 33: 21524–21538.

Belakaria, S.; Deshwal, A.; and Doppa, J. R. 2019. Max-value entropy search for multi-objective Bayesian optimization. *Advances in neural information processing systems*, 32.

Belakaria, S.; Deshwal, A.; Jayakodi, N. K.; and Doppa, J. R. 2020. Uncertainty-aware search framework for multi-objective Bayesian optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 10044–10052.

Burges, C.; Shaked, T.; Renshaw, E.; Lazier, A.; Deeds, M.; Hamilton, N.; and Hullender, G. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, 89–96.

Burges, C. J. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11(23-581): 81.

Chong, S.; Lee, S.; Kim, B.; and Kim, J. 2020. Applications of machine learning in metal-organic frameworks. *Coordination Chemistry Reviews*, 423: 213–487.

Deshwal, A.; Ament, S.; Balandat, M.; Bakshy, E.; Doppa, J. R.; and Eriksson, D. 2023. Bayesian optimization over high-dimensional combinatorial spaces via dictionary-based embeddings. In *International Conference on Artificial Intelligence and Statistics*, 7021–7039. PMLR.

Deshwal, A.; Belakaria, S.; and Doppa, J. R. 2021a. Bayesian optimization over hybrid spaces. In *International conference on machine learning*, 2632–2643. PMLR.

Deshwal, A.; Belakaria, S.; and Doppa, J. R. 2021b. Mercer features for efficient combinatorial Bayesian optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 7210–7218.

Deshwal, A.; Belakaria, S.; Doppa, J. R.; and Kim, D. H. 2022. Bayesian optimization over permutation spaces. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, 6515–6523.

Deshwal, A.; and Doppa, J. 2021. Combining latent space and structured kernels for Bayesian optimization over combinatorial spaces. *Advances in neural information processing systems*, 34: 8185–8200.

Deshwal, A.; Simon, C. M.; and Doppa, J. R. 2021. Bayesian optimization of nanoporous materials. *Molecular Systems Design and Engineering*, 6: 1066–1086.

Doppa, J. R.; Fern, A.; and Tadepalli, P. 2014a. HC-Search: A learning framework for search-based structured prediction. *Journal of Artificial Intelligence Research*, 50: 369–407.

Doppa, J. R.; Fern, A.; and Tadepalli, P. 2014b. Structured prediction via output space search. *The Journal of Machine Learning Research*, 15(1): 1317–1350.

Duda, R. O.; Hart, P. E.; et al. 2006. *Pattern classification*. John Wiley & Sons.

Eriksson, D.; Pearce, M.; Gardner, J.; Turner, R. D.; and Poloczek, M. 2019. Scalable global optimization via local Bayesian optimization. *Advances in neural information processing systems*, 32.

Feng, X.; Ding, X.; and Jiang, D. 2012. Covalent organic frameworks. *Chemical Society Reviews*, 41(18): 6010–6022.

Furukawa, H.; Cordova, K. E.; O'Keeffe, M.; and Yaghi, O. M. 2013. The chemistry and applications of metal-organic frameworks. *Science*, 341(6149): 1230444.

Gantzler, N.; Deshwal, A.; Doppa, J. R.; and Simon, C. M. 2023. Multi-fidelity Bayesian optimization of covalent organic frameworks for xenon/krypton separations. *Digital Discovery*, 2(6): 1937–1956.

Garnett, R. 2023. *Bayesian optimization*. Cambridge University Press.

Hansen, N.; and Ostermeier, A. 2001. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2): 159–195.

Herbrich, R.; Graepel, T.; and Obermayer, K. 2000. Large margin rank boundaries for ordinal regression.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Iserles, A. 2009. *A first course in the numerical analysis of differential equations*. 44. Cambridge university press.

Joachims, T. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 133–142.

Mercado, R.; Fu, R.-S.; Yakutovich, A. V.; Talirz, L.; Haranczyk, M.; and Smit, B. 2018. In silico design of 2D and 3D covalent organic frameworks for methane storage applications. *Chemistry of Materials*, 30(15): 5069–5086.

Moosavi, S. M.; Jablonka, K. M.; and Smit, B. 2020. The Role of Machine Learning in the Understanding and Design of Materials. *Journal of the American Chemical Society*, 142(48): 20273–20287.

Mukherjee, K.; and Colón, Y. J. 2021. Machine learning and descriptor selection for the computational discovery of metal-organic frameworks. *Molecular Simulation*, 1–21.

Suh, M. P.; Park, H. J.; Prasad, T. K.; and Lim, D.-W. 2012. Hydrogen storage in metal–organic frameworks. *Chemical reviews*, 112(2): 782–835.

Sumida, K.; Rogow, D. L.; Mason, J. A.; McDonald, T. M.; Bloch, E. D.; Herm, Z. R.; Bae, T.-H.; and Long, J. R. 2012. Carbon dioxide capture in metal–organic frameworks. *Chemical reviews*, 112(2): 724–781.

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, 3319–3328. PMLR.

Szymanski, N. J.; and Bartel, C. J. 2025. Establishing baselines for generative discovery of inorganic crystals. *Materials Horizons*.

Trickett, C. A.; Helal, A.; Al-Maythalony, B. A.; Yamani, Z. H.; Cordova, K. E.; and Yaghi, O. M. 2017. The chemistry of metal–organic frameworks for CO2 capture, regeneration and conversion. *Nature Reviews Materials*, 2(8): 1–16.

Yaghi, O. M. 2019. Reticular chemistry in all dimensions.

Yang, X.; Ullah, Z.; Stoddart, J. F.; and Yavuz, C. T. 2023. Porous organic cages. *Chemical Reviews*, 123(8): 4602–4634.

Yuan, H.; Li, N.; Fan, W.; Cai, H.; and Zhao, D. 2022. Metal-organic framework based gas sensors. *Advanced Science*, 9(6): 2104374.