Electronic Supplementary Material (ESI) for Digital Discovery.

Supplementary Information: A Multiobjective Closed-loop Approach Towards Autonomous Discovery of Electrocatalysts for Nitrogen Reduction

Lance Kavalsky,[1] Vinay I. Hegde,[2] Bryce Meredig,[2] and Venkatasubramanian Viswanathan[1,*]

[1]*Carnegie Mellon University, Pittsburgh, PA 15213*

[2]*Citrine Informatics, Redwood City, CA 94063*

I. TABULATED PARTIAL SCORES FOR CANDIDATES

Rank	SAA Candidate	Ranking Score	c_j^{active}	C_j	S_j
1	Zr_1Cr	0.38	0.70	0.59	0.91
2	Hf_1Cr	0.35	0.76	0.59	0.78
3	Au_1Re	0.27	0.62	0.68	0.65
4	Ti_1Fe	0.24	0.57	0.89	0.47
5	Ta_1Re	0.19	0.50	0.67	0.57
6	Ru_1Cr	0.10	0.33	0.57	0.55

TABLE I. SAA candidates identified through the multi-objective search with predicted activities that fall within the target window. Raw values of each of the three key metrics are given alongside the target values

II. DATA AND SCRIPTS FOR REPRODUCIBILITY

All data and Python scripts required to perform the analysis presented in this work are made available via the GitHub repository at.

The repository is organized as follows:

1. data/

   - `acsl.json`:

     `autocat.learning.sequential.SequentialLearner` object containing all historical data from the sequential learning search. This may be read using the `SequentialLearner.from_json` method.

   - `acds.json`:

     `autocat.learning.sequential.DesignSpace` object containing all structures within the design space (with calculated labels where available). This may be read using the `DesignSpace.from_json` method.

* venkvis@cmu.edu

- `dft_data.db`:

  `ase.db` containing all of the generated DFT data from the search with entries in the Physical Information File (PIF) format. This may be read using `ase.db.connect` using `type="json"`

- `ELEMENTS.json`:

  `json` containing all chemical species considered in this study

- `raw_volc_m_b.csv`:

  Slopes and intercepts to reproduce the used activity volcano from "The challenge of electrochemical ammonia synthesis: a new perspective on the role of nitrogen scaling relations", Montoya et al., *ChemSusChem* **8** (13), 2180-2186 (2015). DOI: [10.1002/cssc.201500322](https://doi.org/10.1002/cssc.201500322)

- bee_ensembles/:

  Directory containing text files with the BEE energy ensembles for each system that was autonomously identified during the search

2. scripts/

- aq_hist_plot/:

  - `get_aq_hist.py`:
    Script for extracting the acquisition scores and prediction uncertainties as a function of sequential learning (SL) iteration into a text file

  - `make_aq_hist_plot.py`:
    Script to generate a plot of candidate acquisition scores and uncertainties against SL iteration.

  If these scripts are run as-is, will reproduce Figure 3b from the paper.

- drivers/

  - `manage_dft_calculations.py`:
    Script for managing high-throughput adsorption energy calculations on a computing cluster using `fireworks`. Will ensure that first the clean slabs are relaxed before placing the adsorbate.

- `reference_energies.json`:
  Tabulated reference energies used to calculate ΔG_{N} from the DFT total energies of the relaxed systems.
- `sl_driver.py`:
  Script for driving the guided candidate selection with SL. Will automatically re-train the machine learning surrogate, re-calculate the acquisition scores, and suggest the next candidate system for evaluation.

- obj_space_hist_plot/
  - `extract_obj_space_hist.py`:
    Extracts the HHI, Segregation Energies, and ΔG_{N} of both the systems in the initial training set as well as candidates as a function of SL iteration into text files.
  - `make_obj_space_hist_plot.py`:
    Script for generating two subplots. First, it will generate a subplot of the activity volcano with candidates. Second, it will generate a subplot of Normalized HHI against Segregation Energy. Both plots will have candidates colored based on SL iteration.

  If these scripts are run as-is, will reproduce Figure 4 from the paper.

- rank_score_plot/
  - `get_ranking.py`:
    Calculates the partial scores (c_j^{active}, A_j, C_j) and total ranking scores (RS_j) for all candidates and extracts the data into a text file
  - `make_ranking_plot.py`:
    Script for generating the ranking plot of the top 5 identified candidates

  If these scripts are run as-is, will reproduce Figure 5 from the paper

- umap_plots/
  - `L1_EMBEDDING.txt`:
    Contains the UMAP embeddings of all systems in the considered SAA design space that were used in the paper.

- `make_umap_plot_initial_only`:
  Script for generating plot of UMAP projection with only the initial training points highlighted (Figure 1d in the paper)
- `make_umap_plot.py`:
  Script for generating plot of UMAP projection with both the initial training points highlighted alongside the identified candidates as a function of iteration (Figure 3a in the paper)
- `umap_calc.py`:
  Calculate UMAP embeddings for the SAA design space using magpie featurization. **N.B.** due to the stochasticity in the UMAP approach, running this script as-is does not guarantee identical embeddings to that provided in `L1_EMBEDDING.txt`, but overall trends should remain

A. Running the scripts

The required packages for executing the scripts are specified in `requirements.txt`, and can be installed in a new environment (e.g. using conda) as follows:

```
$ conda create -n multi_obj_search python=3.10
$ conda activate multi_obj_search
$ pip install -r requirements.txt
```

The scripts are all in python, and can be run from the command line. For example:

```
$ cd scripts/aq_hist_plot
$ python get_aq_hist.py
```

III. COMPARISON WITH A THRESHOLDS-BASED FILTERING APPROACH

Here, we analyze a common alternate strategy that uses target threshold values to filter out candidates, instead of an acquisition function that uses an aggregate score (based on ML predictions and uncertainties) to rank and select candidates sequentially (as presented in the main text), and its effect on the final top-ranked candidates identified. For this, we leverage the tabulated HHI and segregation energy data, as well as the DFT-calculated adsorption

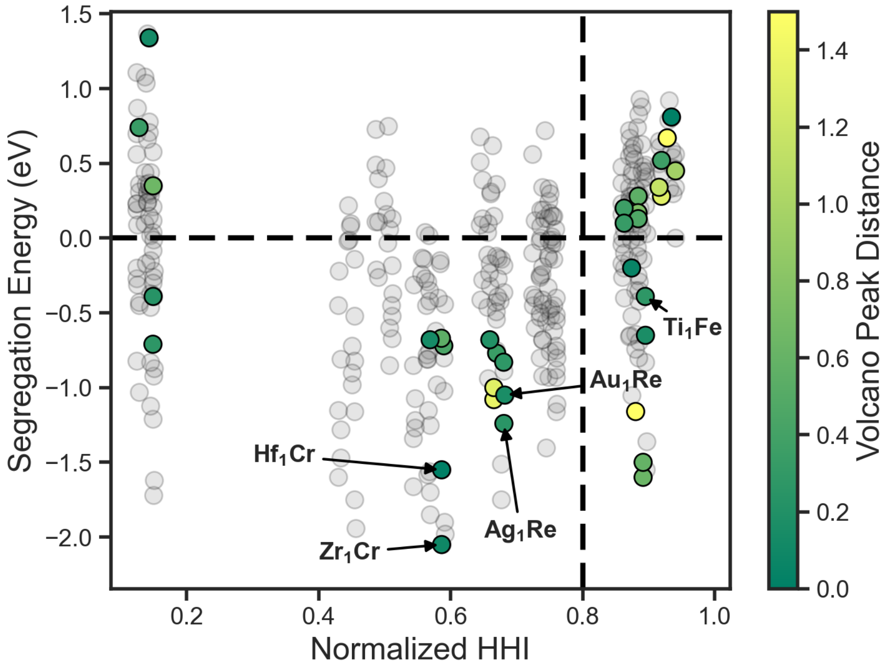

FIG. S1. Normalized HHI and Segregation for entire SAA design space. Colored points, visualizing indicating predicted distance from the volcano peak, are systems evaluated with DFT, either as part of the initial training set or evaluated during the search campaign. The top 5 ranked candidates are explicitly labelled. Dashed lines are arbitrary filtering thresholds used for demonstration purposes. 4 out of the 5 top-ranked candidates would have been missed if these filters were applied before the search campaign.

energies computed in this work. FIG. S1 shows the normalized HHI and segregation energies for the full SAA design space, as well as distances from the activity volcano peak (for evaluated systems). Here we consider two filtering thresholds. Along the segregation energy axis we filter out systems above 0 eV as these host-dopant combinations are predicted to thermodynamically drive the dopant into the bulk (not preferred). While this threshold of 0 eV is a natural choice for segregation energy, for other metrics, the selection of a threshold can be highly non-trivial and rely on extensive domain knowledge. For example, the choice of a threshold value for the HHI axis is not obvious. For demonstration purposes here, we filter out systems with HHI below 0.8 (based on the the visual clusters in the data), but a reasonable argument can be made for a different value based on the cost targets for a particular application.

From the combination of the above two filters, 4 out of the top 5 ranked candidate systems previously identified from our search are filtered out. While this does not rule out the possibility of finding high-quality candidates within the confined region defined by the target threshold values, it highlights how pre-defined threshold choices can lead to missing systems that are promising overall. SL frameworks that use scores based on ML predictions *and* uncertainties to rank and select candidates are not similarly affected.