Data in Brief

Materials Science Optimization Benchmark Dataset for Multi-Objective, Multi-Fidelity Optimization of Hard-Sphere Packing Simulations

--Manuscript Draft--

Manuscript Number:	DIB-D-23-00033R1
Article Type:	Data Article
Keywords:	adaptive design; physics-based; Lubachevsky-Stillinger; force-biased algorithms; Machine learning; packing generation; Transfer Learning; matsci-opt-benchmarks
Corresponding Author:	Sterling Gregory Baird, M.Sc. The University of Utah Salt Lake City, Utah UNITED STATES
First Author:	Sterling Gregory Baird, M.Sc.
Order of Authors:	Sterling Gregory Baird, M.Sc.
	Ramsey Issa
	Taylor D. Sparks, Ph.D.
Abstract:	In scientific disciplines, benchmarks play a vital role in driving progress forward. For a benchmark to be effective, it must closely resemble real-world tasks. If the level of difficulty or relevance is inadequate, it can impede progress in the field. Moreover, benchmarks should have low computational overhead to ensure accessibility and repeatability. The objective is to achieve a kind of "Turing test" by creating a surrogate model that is practically indistinguishable from the ground truth observation, at least within the dataset's explored boundaries. This objective necessitates a large quantity of data. This data encompasses numerous features that are characteristic of chemistry and materials science optimization tasks that are relevant to industry. These features include high levels of noise, multiple fidelities, multiple objectives, linear constraints, non-linear correlations, and failure regions. We performed 494498 random hard-sphere packing simulations representing 206 CPU days' worth of computational overhead. Simulations required nine input parameters with linear constraints and two discrete fidelities each with continuous fidelity parameters. The data was logged in a free-tier shared MongoDB Atlas database, producing two core tabular datasets: a failure probability dataset and a regression dataset. The failure probability dataset maps unique input parameter sets to the estimated probabilities that the simulation will fail. The regression dataset maps input parameter sets (including repeats) to particle packing fractions and computational runtimes for each of the two steps. These two datasets were used to create a surrogate model as close as possible to running the actual simulations by incorporating simulation failure and heteroskedastic noise. In the regression dataset, percentile ranks were calculated for each group of identical parameter sets to account for heteroskedastic noise, thereby ensuring reliable and accurate data. This differs from the conventional approach that imposes a-priori assumptions, such as Gaussian noise, by specifying mean and standard deviation. This technique can be extended to other benchmark datasets to bridge the gap between optimization benchmarks with low computational overhead and the complex optimization scenarios encountered in the real world.

Cover Letter

Materials Science and Engineering

122 S. Central Campus Drive, Salt Lake City, Utah 84112 (801) 581-8632

March 3, 2023

Dear Editor:

We are very pleased to submit this data article to *Data in Brief*. Our manuscript, titled *Materials Science Optimization Benchmark Dataset for Multi-Objective, Multi-Fidelity Optimization of Hard-Sphere Packing Simulations* provides important research from my research group at the University of Utah. This work presents a benchmark dataset for materials science optimization tasks that incorporates both simulation failure and heteroskedastic noise in a realistically complex setting.

The dataset represents 206 days' worth of CPU computation time and contains over 400,000 datapoints. The two datasets presented in this work are used to create a surrogate model as close as possible to running the actual simulations. This will help form part of a larger suite of experimentally and computationally derived benchmarks. Additionally, this dataset can serve as an optimization task for advanced Bayesian optimization topics including multi-fidelity and linearly constrained optimization.

Sincerely,

Dr. Taylor Sparks
Associate Professor & Associate Chair
Materials Science and Engineering Department
University of Utah
Salt Lake City, Utah 84112

Response to Reviewers

Reviewer #1

Thank you for your comments and suggestions! We have addressed each of your comments below.

Page 2

Description of data collection

> I feel that one can give a more gentle introduction to an underlying problem of packing simulation here.

Mentioned that we use geometry-based particle packing simulations to generate a set of spheres within a volume and analyze the packing fraction (occupied space vs. total space).

> One can also explain here or below why exactly simulations fail. Is it an underlying property of the algorithm/simulation or of this particular realization that you used? For example, https://github.com/VasiliBaranov/packing-generation uses for the LS algorithm the Verlet lists/cell lists optimization to find relevant particle neighbors quickly, but for very dilute packings it can miss some relevant neighbors, so the algorithm can fail due to missing some relevant neighbors during approximate search. In this case, it is the consequence of implementation.

Mentioned that the algorithm can fail due to missing neighboring particles during an approximate search.

Page 3

> > The failure probability dataset (sobol_probability_filter.csv) contains unique input parameter sets (nine variables) and the estimated probabilities that the simulation will fail at each of the two steps (force-biased algorithm and Lubachevsky-Stillinger).
>
> It would be nice to put links to the force-biased algorithm and Lubachevsky-Stillinger papers here.

References have been added here.

Page 4

> > Figure 1 contains a histogram for the number of successful repeats for each parameter combination.

> It took me some time to decipher this sentence and figure.

This has been updated for clarity, including an example.

> > Figure 2 contains the probability of a simulation failing for each of the two algorithms.
>
> I am not sure that I understand it. If this is the probability of failure for two algorithms, there shall be two numbers..

This has been changed to: "For a given parameter set, the probability of a simulation failing is the number of failed simulations divided by the number of simulations that were run. Figure 2 contains the probabilities of a parameter set failing for each of the two algorithms (force-biased and Lubachevsky–Stillinger)."

Page 6

> > Figure 2. Histogram of normalized simulation counts vs. the probability of a simulation failing for a given parameter set.
>
> Ah, so for each parameter set you track how many simulations you did and what was the corresponding probability of failure? Maybe you can mention this or explain better somehow.

That's correct. This has been updated per the comment above.

Page 7

> > Figure 3. Histogram of number of simulations vs. packing fraction for the force-biased algorithm or fba (blue) and Lubachevsky-Stillinger or ls algorithm (red). On average, the ls algorithm tends to have higher packing fractions with a more Gaussian-like distribution than fba.
>
> It is a bit suspicious that the LS algorithm produces packings only above density 0.64, i.e. jammed packings only. Maybe you used the version of the algorithm that is called lsgd in https://github.com/VasiliBaranov/packing-generation It is then slightly different from the pure LS algorithm (the compression rate decreases) and it can be explained a little in the text.

We have updated this to reflect that the algorithms were run in a two-step process: first an attempt to run FBA followed by an attempt to run LS. Even if FBA failed, LS was still attempted. We included a link to the relevant code.

> In general, one could explain the underlying problem here slightly more and may structure the section "Experimental design, materials and methods" to make it easier to follow, I feel.

This has been addressed and the structure updated.

> Also, it would make sense to mention https://github.com/VasiliBaranov/packing-generation in section Experimental design, materials and methods.

This has been added.

> It would also be fair to cite the papers where the code above was introduced and used, e.g. https://pubs.rsc.org/en/content/articlehtml/2014/sm/c3sm52959b

This has been referenced.

> > We acknowledge Vasili Baranov and Robin De Schepper for help with the packing-generation codebase
>
> It is probably more appropriate to use in acknowledgements the author name from the paper, https://pubs.rsc.org/en/content/articlehtml/2014/sm/c3sm52959b, Vasili Baranau.

This has been updated.

Article information

Article title

Materials Science Optimization Benchmark Dataset for Multi-Objective, Multi-Fidelity Optimization of Hard-Sphere Packing Simulations

Authors

Sterling G. Baird[1]*, Ramsey Issa[1], Taylor D. Sparks[1,2]

Affiliations

1. Materials Science & Engineering, 122 S. Central Campus Drive, #304 Salt Lake City, Utah 84112-0056
2. Chemistry Department, University of Liverpool, Liverpool, L7 3NY, United Kingdom

Corresponding author’s email address and Twitter handle

sterling.baird@utah.edu
@SterlingBaird1

Keywords

adaptive design, physics-based, Lubachevsky–Stillinger, force-biased algorithms, particle packing, packing generation, transfer learning, size distribution

Abstract

In scientific disciplines, benchmarks play a vital role in driving progress forward. For a benchmark to be effective, it must closely resemble real-world tasks. If the level of difficulty or relevance is inadequate, it can impede progress in the field. Moreover, benchmarks should have low computational overhead to ensure accessibility and repeatability. The objective is to achieve a kind of "Turing test" by creating a surrogate model that is practically indistinguishable from the ground truth observation, at least within the dataset's explored boundaries. This objective necessitates a large quantity of data. This data encompasses numerous features that are characteristic of chemistry and materials science optimization tasks that are relevant to industry. These features include high levels of noise, multiple fidelities, multiple objectives, linear constraints, non-linear correlations, and failure regions. We performed 494498 random hard-sphere packing simulations representing 206 CPU days’ worth of computational overhead. Simulations required nine input parameters with linear constraints and two discrete fidelities each with continuous fidelity parameters. The data was logged in a free-tier shared MongoDB Atlas database, producing two core tabular datasets: a failure probability dataset and a regression dataset. The failure probability dataset maps unique input parameter sets to the estimated probabilities that the simulation will fail. The regression dataset maps input parameter sets (including repeats) to particle packing fractions and computational runtimes for each of the two steps. These two datasets were used to create a surrogate model as close as possible to

running the actual simulations by incorporating simulation failure and heteroskedastic noise. In the regression dataset, percentile ranks were calculated for each group of identical parameter sets to account for heteroskedastic noise, thereby ensuring reliable and accurate data. This differs from the conventional approach that imposes a-priori assumptions, such as Gaussian noise, by specifying mean and standard deviation. This technique can be extended to other benchmark datasets to bridge the gap between optimization benchmarks with low computational overhead and the complex optimization scenarios encountered in the real world.

Specifications table

Subject	Computational materials science
Specific subject area	Physics-based geometric packing
Type of data	Table Figure
How the data were acquired	Data was acquired by running compiled C software hosted at https://github.com/VasiliBaranov/packing-generation in a two-step process orchestrated using Python in https://github.com/sparks-baird/matsci-opt-benchmarks/blob/main/scripts/particle_packing/packing_generation_submitit.py. The Python code was utilized as a driver for the compiled packing generation executable and executed using the resources provided by the University of Utah's Center for High-performance Computing (CHPC). The submission of jobs to the SLURM scheduler was facilitated through https://github.com/facebookincubator/submitit, and the MongoDB Data API was utilized to record data in JSON format. For a snapshot of the code utilized in matsci-opt-benchmarks, please refer to https://github.com/sparks-baird/matsci-opt-benchmarks/tree/v0.2.2 (https://zenodo.org/record/7697264#.ZAJo6nbMIeM).
Data format	Raw Analyzed Filtered
Description of data collection	We use geometry-based particle packing simulations to generate a set of spheres within a volume and analyze the packing fraction (occupied space vs. total space). A total of 65536 parameter combinations were randomly sampled using quasi-random Sobol sampling, varying seven irreducible parameters in addition to the number of particles and initial scaling factor. A constrained search

	space was employed through the Ax Platform with repeats. Out of these simulations, 494498 were successfully completed, requiring 206 CPU days to run. Packing simulations were run using two algorithms run sequentially (i.e., a two-step process). Sometimes, the algorithms can fail. For example, during an approximate search of neighboring particles, sometimes not all neighboring particles are found. Failed simulations were recorded as NaN values with ratio of successful to total simulations tracked on a per parameter set basis (sobol_probability_filter.csv). Repeat simulations were grouped and ranked by percentile using the “dense” method with pct=True in pandas.core.groupby.GroupBy.rank (sobol_regression.csv)[1]. Surrogate models were fitted for failure probability, packing fraction, and computational runtime for each of two particle packing algorithms, totaling six surrogate models.
Data source location	University of Utah, Salt Lake City UT USA
Data accessibility	Repository name: Zenodo Data identification number: 7696165 Direct URL to data: https://dx.doi.org/10.5281/zenodo.7696165

Value of the data

- Valuable for adaptive design benchmarking
- Benefits optimization researchers and practitioners in the physical sciences
- Provides insight into packing behavior in powder-bed additive manufacturing, can be integrated with experimental data
- Provides an example for future datasets

Objective

Optimization tasks that are relevant to industry in the fields of materials science and chemistry are typically hierarchical, noisy, multi-fidelity[2,3], multi-objective[4,5], high-dimensional[6,7], non-linearly correlated, and involve mixed numerical and categorical variables subject to linear[8] and non-linear constraints. Existing benchmark datasets[9–14] have limitations as they ignore or simplify the impact of noise and the occurrence of failure with certain parameter combinations. By integrating simulation failure and heteroskedastic noise, we aim to achieve a "Turing test" scenario where the surrogate model is practically indistinguishable from the ground truth simulation. This strategy bridges the gap between low-cost surrogate functions based on benchmark datasets and the high-cost evaluation of objective functions in real-world scenarios.

Data description

The failure probability dataset (sobol_probability_filter.csv) contains unique input parameter sets (nine variables) and the estimated probabilities that the simulation will fail at each of the two steps (force-biased algorithm[15,16] and Lubachevsky–Stillinger[17–19]).

The regression dataset (sobol_regression.csv) contains input parameters (including repeats) spanning nine variables and corresponding particle packing fractions as well as computational runtimes for each of the two steps (force-biased algorithm and Lubachevsky–Stillinger).

There are six regression models (surrogate_models.pkl) trained on all data meant for production use. These six models can be used together to create the benchmark function.

There are five cross-validation sets of six regression models (cross_validation_models_0.pkl, cross_validation_models_1.pkl, cross_validation_models_2.pkl, cross_validation_models_3.pkl, cross_validation_models_4.pkl).

The model metadata (model_metadata.json) contains the raw mean absolute error scores, the raw predictions, and the true values for each of the cross-validation folds.

For each group of repeats, we tracked the number of simulations that were run and the number of simulations that ran successfully. Figure 1 contains a histogram of ~~for~~ the number of successful repeats for each parameter combination. For example, of the 65536 unique parameter combinations, approximately 5000 had eight successful repeats.

For a given parameter set, the probability of a simulation failing is the number of failed simulations divided by the number of simulations that were run. Figure 2 contains the probabilities~~y~~ of a parameter set ~~simulation~~ failing for~~-~~ each of the two algorithms (force-biased and Lubachevsky–Stillinger)~~each of the two algorithms~~.

Figure 3 contains the histograms of observed particle packing fractions for each of the two algorithms.

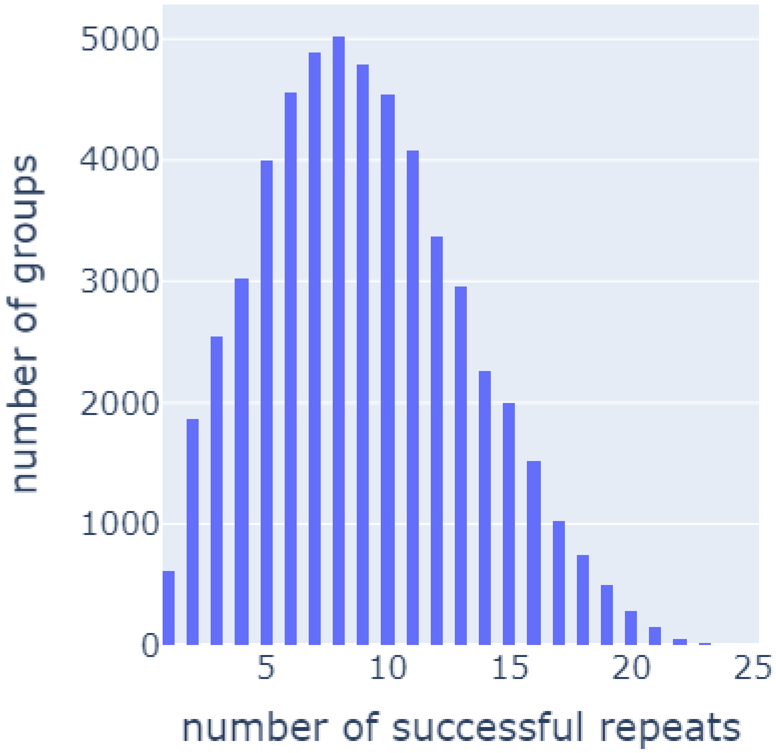

Figure 1. Histogram of number of parameter groups vs. number of successful repeats within a given group. For example, of the 65536 unique parameter combinations, approximately 5000 had eight successful repeats.

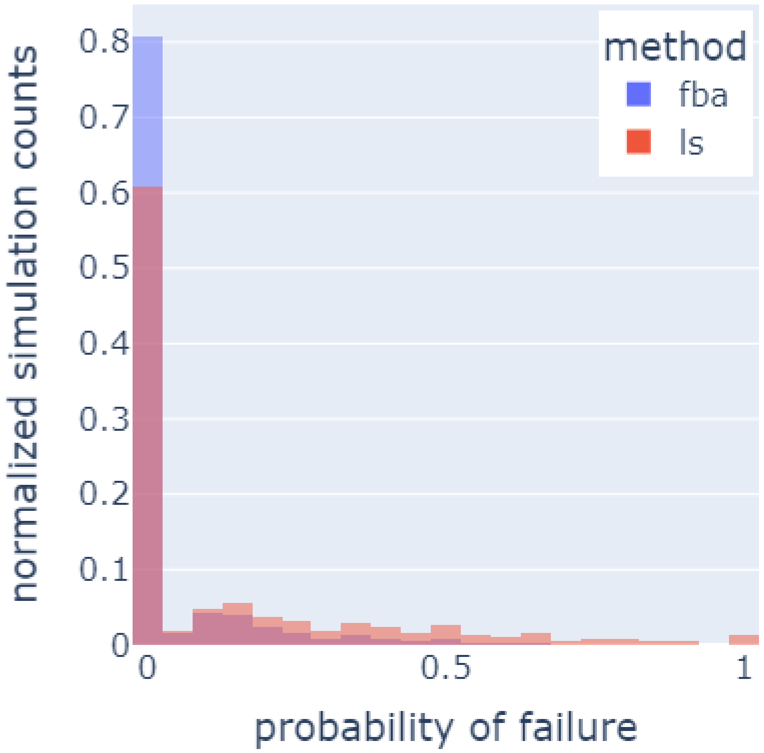

Figure 2. Histogram of normalized simulation counts vs. the probability of a simulation failing for a given parameter set. On average, the force-biased algorithm or fba (blue) is more likely to succeed than the Lubachevsky–Stillinger or ls (red) algorithm.

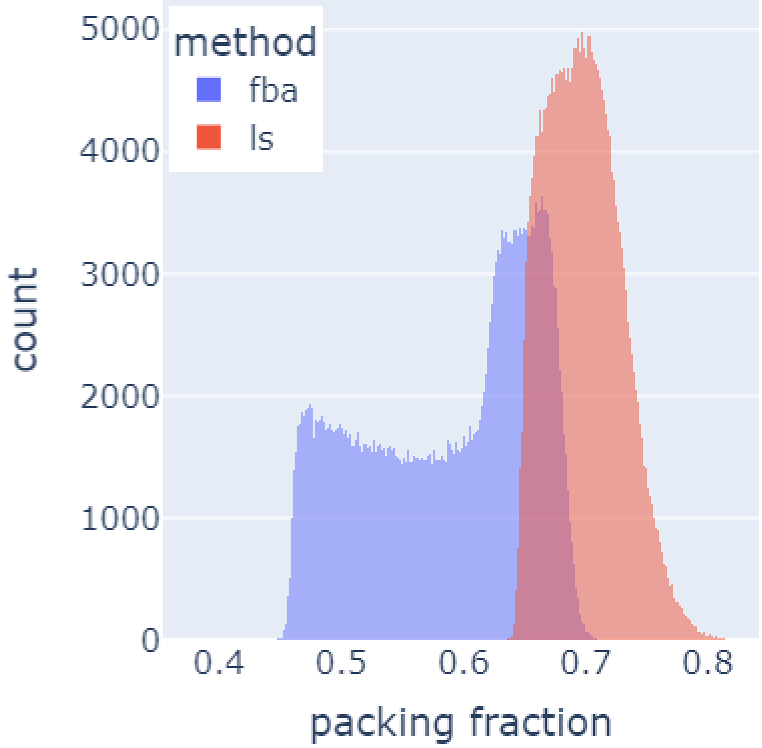

Figure 3. Histogram of number of simulations vs. packing fraction for the force-biased algorithm or fba (blue) and Lubachevsky–Stillinger or ls algorithm (red). On average, the ls algorithm tends to have higher packing fractions with a more Gaussian-like distribution than fba.

Experimental design, materials and methods

For this dataset, we aim to achieve a “Turing test” scenario where the surrogate model for a simulation is practically indistinguishable from the corresponding ground truth simulation. Here, we use https://github.com/VasiliBaranov/packing-generation[20] to run hard-sphere particle packing simulations while varying the particle size distribution. We ran repeat simulations to better capture noise, and we also tracked when simulations fail and the computational runtime at each step. ~~In this dataset, 494498 hard-sphere packing simulations were recorded using a~~ Particle packing simulations were performed in a two-step process of a force-biased algorithm[15,16] followed by the Lubachevsky–Stillinger algorithm[17–19]. An attempt to run the LS algorithm was always preceded by an attempt to run the FBA algorithm. If the force-biased algorithm failed, the Lubachevsky–Stillinger algorithm was still attempted (https://github.com/sparks-baird/matsci-opt-benchmarks/blob/v0.2.2/src/matsci_opt_benchmarks/particle_packing/utils/packing_generation.py#L63-L183). The simulations were performed using mixtures of three different particle types, each characterized by two log-normal distribution parameters and three composition parameters. Two parameters (scale and shape) describe each of the three distributions, and three additional composition parameters describe the fractional share (e.g., in terms of volume) of each of the particle types. These nine parameters fully define the particle size distribution. With appropriate constraints applied, only seven of these parameters are necessary to fully define the particle size distribution. Additionally, the number of particles and an initial scaling factor were allowed to vary. With a greater number of particles, denser and more realistic packs can be generated at the expense of computational cost (i.e., the fidelity parameter). The initial scaling factor affects the computational stability of the simulation; with an adequate scaling factor, the simulation is more likely to be completed successfully. The ~~The~~ quasi-random Sobol sampling technique was employed to generate parameter combinations, enabling a more uniform sampling of the allowable parameter space. We sampled 65536 unique parameter combinations. Repeat simulations for the parameter combinations were run to capture heteroskedastic noise, totaling 494498 simulations. ~~Although it may serve other purposes, this dataset was primarily designed as a multi-fidelity benchmark dataset for constrained adaptive design experiments.~~ ~~To realistically capture the noise in this dataset, simulations were run multiple times for each quasi-random parameter combination.~~ To increase throughput and reduce latency, simulation parameters (including repeats) were shuffled and divided into batches, which were then dispatched to a high-performance computing environment for asynchronous evaluation. The data were recorded in a free-tier MongoDB Atlas database and then consolidated and prepared as datasets suitable for machine learning applications. Although it may serve other purposes, this dataset was primarily designed as a multi-fidelity benchmark dataset for constrained adaptive design experiments, hence the tracking of repeats, running simulations at various fidelities, incorporation of constraints, and tracking when simulations fail and the computational expense (whether or not the simulation runs successfully). For further implementation details, see https://github.com/sparks-baird/matsci-opt-benchmarks/tree/v0.2.2/scripts/particle_packing and https://github.com/sparks-baird/matsci-opt-benchmarks/tree/v0.2.2/notebooks/particle_packing. Instructions for model usage are available at https://matsci-opt-benchmarks.readthedocs.io/.

Ethics statements

There are no statements to declare.

CRediT author statement

Sterling G. Baird: Project administration, Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization. **Ramsey Issa**: Methodology, Software, Validation, Formal Analysis, Writing - Review & Editing. **Taylor D. Sparks**: Supervision, Funding acquisition

Acknowledgments

Funding: This work was supported by the National Science Foundation Division of Materials Research [Grant no. DMR-1651668].

We acknowledge Vasili Baranau~~ov~~ and Robin De Schepper for help with the packing-generation codebase. We acknowledge Ramsey Issa and Hasan M. Sayeed for reviewing the manuscript and providing valuable feedback. We acknowledge Jason R. Hall for discussion of particle packing simulations in the context of industry processes. We acknowledge the University of Utah's Center for High Performance Computing (CHPC) for providing computational resources. We acknowledge OpenAI for providing free usage of their research tool, ChatGPT, which was used during the writing, review, and editing process.

Declaration of interests

x The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

References

(1) Baird, S. G. Materials Science Optimization Benchmark Dataset for Multi-Fidelity Hard-Sphere Packing Simulations, 2023. https://doi.org/10.5281/zenodo.7513019.

(2) Ghoreishi, S. F.; Molkeri, A.; Arróyave, R.; Allaire, D.; Srivastava, A. Efficient Use of Multiple Information Sources in Material Design. *Acta Materialia* **2019**, *180*, 260–271. https://doi.org/10.1016/j.actamat.2019.09.009.

(3) Kandasamy, K.; Vysyaraju, K. R.; Neiswanger, W.; Paria, B.; Collins, C. R.; Schneider, J.; Poczos, B.; Xing, E. P. Tuning Hyperparameters without Grad Students: Scalable and Robust Bayesian Optimisation with Dragonfly. *arXiv:1903.06694 [cs, stat]* **2020**.

(4) Hanaoka, K. Comparison of Conceptually Different Multi-Objective Bayesian Optimization Methods for Material Design Problems. *Materials Today Communications* **2022**, 103440. https://doi.org/10.1016/j.mtcomm.2022.103440.

(5) Häse, F.; Roch, L. M.; Aspuru-Guzik, A. Chimera: Enabling Hierarchy Based Multi-Objective Optimization for Self-Driving Laboratories. *Chem. Sci.* **2018**, *9* (39), 7642–7655. https://doi.org/10.1039/C8SC02239A.

(6) Baird, S. G.; Liu, M.; Sparks, T. D. High-Dimensional Bayesian Optimization of 23 Hyperparameters over 100 Iterations for an Attention-Based Network to Predict Materials Property: A Case Study on CrabNet Using Ax Platform and SAASBO. *Computational Materials Science* **2022**, *211*, 111505. https://doi.org/10.1016/j.commatsci.2022.111505.

(7) Eriksson, D.; Jankowiak, M. High-Dimensional Bayesian Optimization with Sparse Axis-Aligned Subspaces. *arXiv:2103.00349 [cs, stat]* **2021**.

(8) Baird, S.; Hall, J. R.; Sparks, T. D. The Most Compact Search Space Is Not Always the Most Efficient: A Case Study on Maximizing Solid Rocket Fuel Packing Fraction via Constrained Bayesian Optimization. ChemRxiv September 6, 2022. https://doi.org/10.26434/chemrxiv-2022-nz2w8-v2.

(9) Dunn, A.; Wang, Q.; Ganose, A.; Dopp, D.; Jain, A. Benchmarking Materials Property Prediction Methods: The Matbench Test Set and Automatminer Reference Algorithm. *npj Comput Mater* **2020**, *6* (1), 138. https://doi.org/10.1038/s41524-020-00406-3.

(10) De Breuck, P.-P.; Evans, M. L.; Rignanese, G.-M. Robust Model Benchmarking and Bias-Imbalance in Data-Driven Materials Science: A Case Study on MODNet. *J. Phys.: Condens. Matter* **2021**, *33* (40), 404002. https://doi.org/10.1088/1361-648X/ac1280.

(11) Wang, A.; Liang, H.; McDannald, A.; Takeuchi, I.; Kusne, A. G. Benchmarking Active Learning Strategies for Materials Optimization and Discovery. arXiv April 12, 2022. http://arxiv.org/abs/2204.05838 (accessed 2022-07-04).

(12) Liang, Q.; Gongora, A. E.; Ren, Z.; Tiihonen, A.; Liu, Z.; Sun, S.; Deneault, J. R.; Bash, D.; Mekki-Berrada, F.; Khan, S. A.; Hippalgaonkar, K.; Maruyama, B.; Brown, K. A.; Fisher III, J.; Buonassisi, T. Benchmarking the Performance of Bayesian Optimization across Multiple Experimental Materials Science Domains. *npj Comput Mater* **2021**, *7* (1), 188. https://doi.org/10.1038/s41524-021-00656-9.

(13) Henderson, A. N.; Kauwe, S. K.; Sparks, T. D. Benchmark Datasets Incorporating Diverse Tasks, Sample Sizes, Material Systems, and Data Heterogeneity for Materials Informatics. *Data in Brief* **2021**, *37*, 107262. https://doi.org/10.1016/j.dib.2021.107262.

(14) Häse, F.; Aldeghi, M.; Hickman, R. J.; Roch, L. M.; Christensen, M.; Liles, E.; Hein, J. E.; Aspuru-Guzik, A. Olympus: A Benchmarking Framework for Noisy Optimization and Experiment Planning. *Mach. Learn.: Sci. Technol.* **2021**, *2* (3), 035021. https://doi.org/10.1088/2632-2153/abedc8.

(15) Mościński, J.; Bargieł, M.; Rycerz, Z. A.; Jacobs, P. W. M. The Force-Biased Algorithm for the Irregular Close Packing of Equal Hard Spheres. *Molecular Simulation* **1989**, *3* (4), 201–212. https://doi.org/10.1080/08927028908031373.

(16) Bezrukov, A.; Bargieł, M.; Stoyan, D. Statistical Analysis of Simulated Random Packings of Spheres. *Particle & Particle Systems Characterization* **2002**, *19* (2), 111–118. https://doi.org/10.1002/1521-4117(200205)19:2<111::AID-PPSC111>3.0.CO;2-M.

(17) Skoge, M.; Donev, A.; Stillinger, F. H.; Torquato, S. Packing Hyperspheres in High-Dimensional Euclidean Spaces. *Phys. Rev. E* **2006**, *74* (4), 041127. https://doi.org/10.1103/PhysRevE.74.041127.

(18) Lubachevsky, B. D. How to Simulate Billiards and Similar Systems. *Journal of Computational Physics* **1991**, *94* (2), 255–283. https://doi.org/10.1016/0021-9991(91)90222-7.

(19) Lubachevsky, B. D.; Stillinger, F. H. Geometric Properties of Random Disk Packings. *Journal of Statistical Physics* **1990**, *60* (5), 561–583. https://doi.org/10.1007/BF01025983.

(20) Baranau, V.; Tallarek, U. Random-Close Packing Limits for Monodisperse and Polydisperse Hard Spheres. *Soft Matter* **2014**, *10* (21), 3826–3841. https://doi.org/10.1039/C3SM52959B.

Article information

Article title

Materials Science Optimization Benchmark Dataset for Multi-Objective, Multi-Fidelity Optimization of Hard-Sphere Packing Simulations

Authors

Sterling G. Baird[1]*, Ramsey Issa[1], Taylor D. Sparks[1,2]

Affiliations

1. Materials Science & Engineering, 122 S. Central Campus Drive, #304 Salt Lake City, Utah 84112-0056
2. Chemistry Department, University of Liverpool, Liverpool, L7 3NY, United Kingdom

Corresponding author's email address and Twitter handle

sterling.baird@utah.edu
@SterlingBaird1

Keywords

adaptive design, physics-based, Lubachevsky–Stillinger, force-biased algorithms, particle packing, packing generation, transfer learning, size distribution

Abstract

In scientific disciplines, benchmarks play a vital role in driving progress forward. For a benchmark to be effective, it must closely resemble real-world tasks. If the level of difficulty or relevance is inadequate, it can impede progress in the field. Moreover, benchmarks should have low computational overhead to ensure accessibility and repeatability. The objective is to achieve a kind of "Turing test" by creating a surrogate model that is practically indistinguishable from the ground truth observation, at least within the dataset's explored boundaries. This objective necessitates a large quantity of data. This data encompasses numerous features that are characteristic of chemistry and materials science optimization tasks that are relevant to industry. These features include high levels of noise, multiple fidelities, multiple objectives, linear constraints, non-linear correlations, and failure regions. We performed 494498 random hard-sphere packing simulations representing 206 CPU days' worth of computational overhead. Simulations required nine input parameters with linear constraints and two discrete fidelities each with continuous fidelity parameters. The data was logged in a free-tier shared MongoDB Atlas database, producing two core tabular datasets: a failure probability dataset and a regression dataset. The failure probability dataset maps unique input parameter sets to the estimated probabilities that the simulation will fail. The regression dataset maps input parameter sets (including repeats) to particle packing fractions and computational runtimes for each of the two steps. These two datasets were used to create a surrogate model as close as possible to

running the actual simulations by incorporating simulation failure and heteroskedastic noise. In the regression dataset, percentile ranks were calculated for each group of identical parameter sets to account for heteroskedastic noise, thereby ensuring reliable and accurate data. This differs from the conventional approach that imposes a-priori assumptions, such as Gaussian noise, by specifying mean and standard deviation. This technique can be extended to other benchmark datasets to bridge the gap between optimization benchmarks with low computational overhead and the complex optimization scenarios encountered in the real world.

Specifications table

Subject	Computational materials science
Specific subject area	Physics-based geometric packing
Type of data	Table Figure
How the data were acquired	Data was acquired by running compiled C software hosted at https://github.com/VasiliBaranov/packing-generation in a two-step process orchestrated using Python in https://github.com/sparks-baird/matsci-opt-benchmarks/blob/main/scripts/particle_packing/packing_generation_submitit.py. The Python code was utilized as a driver for the compiled packing generation executable and executed using the resources provided by the University of Utah's Center for High-performance Computing (CHPC). The submission of jobs to the SLURM scheduler was facilitated through https://github.com/facebookincubator/submitit, and the MongoDB Data API was utilized to record data in JSON format. For a snapshot of the code utilized in matsci-opt-benchmarks, please refer to https://github.com/sparks-baird/matsci-opt-benchmarks/tree/v0.2.2 (https://zenodo.org/record/7697264#.ZAJo6nbMIeM).
Data format	Raw Analyzed Filtered
Description of data collection	We use geometry-based particle packing simulations to generate a set of spheres within a volume and analyze the packing fraction (occupied space vs. total space). A total of 65536 parameter combinations were randomly sampled using quasi-random Sobol sampling, varying seven irreducible parameters in addition to the number of particles and initial scaling factor. A constrained search

	space was employed through the Ax Platform with repeats. Out of these simulations, 494498 were successfully completed, requiring 206 CPU days to run. Packing simulations were run using two algorithms run sequentially (i.e., a two-step process). Sometimes, the algorithms can fail. For example, during an approximate search of neighboring particles, sometimes not all neighboring particles are found. Failed simulations were recorded as NaN values with ratio of successful to total simulations tracked on a per parameter set basis (sobol_probability_filter.csv). Repeat simulations were grouped and ranked by percentile using the “dense” method with pct=True in pandas.core.groupby.GroupBy.rank (sobol_regression.csv)[1]. Surrogate models were fitted for failure probability, packing fraction, and computational runtime for each of two particle packing algorithms, totaling six surrogate models.
Data source location	University of Utah, Salt Lake City UT USA
Data accessibility	Repository name: Zenodo Data identification number: 7696165 Direct URL to data: https://dx.doi.org/10.5281/zenodo.7696165

Value of the data

- Valuable for adaptive design benchmarking
- Benefits optimization researchers and practitioners in the physical sciences
- Provides insight into packing behavior in powder-bed additive manufacturing, can be integrated with experimental data
- Provides an example for future datasets

Objective

Optimization tasks that are relevant to industry in the fields of materials science and chemistry are typically hierarchical, noisy, multi-fidelity[2,3], multi-objective[4,5], high-dimensional[6,7], non-linearly correlated, and involve mixed numerical and categorical variables subject to linear[8] and non-linear constraints. Existing benchmark datasets[9–14] have limitations as they ignore or simplify the impact of noise and the occurrence of failure with certain parameter combinations. By integrating simulation failure and heteroskedastic noise, we aim to achieve a "Turing test" scenario where the surrogate model is practically indistinguishable from the ground truth simulation. This strategy bridges the gap between low-cost surrogate functions based on benchmark datasets and the high-cost evaluation of objective functions in real-world scenarios.

Data description

The failure probability dataset (sobol_probability_filter.csv) contains unique input parameter sets (nine variables) and the estimated probabilities that the simulation will fail at each of the two steps (force-biased algorithm[15,16] and Lubachevsky–Stillinger[17–19]).

The regression dataset (sobol_regression.csv) contains input parameters (including repeats) spanning nine variables and corresponding particle packing fractions as well as computational runtimes for each of the two steps (force-biased algorithm and Lubachevsky–Stillinger).

There are six regression models (surrogate_models.pkl) trained on all data meant for production use. These six models can be used together to create the benchmark function.

There are five cross-validation sets of six regression models (cross_validation_models_0.pkl, cross_validation_models_1.pkl, cross_validation_models_2.pkl, cross_validation_models_3.pkl, cross_validation_models_4.pkl).

The model metadata (model_metadata.json) contains the raw mean absolute error scores, the raw predictions, and the true values for each of the cross-validation folds.

For each group of repeats, we tracked the number of simulations that were run and the number of simulations that ran successfully. Figure 1 contains a histogram of the number of successful repeats for each parameter combination. For example, of the 65536 unique parameter combinations, approximately 5000 had eight successful repeats.

For a given parameter set, the probability of a simulation failing is the number of failed simulations divided by the number of simulations that were run. Figure 2 contains the probabilities of a parameter set failing for each of the two algorithms (force-biased and Lubachevsky–Stillinger).

Figure 3 contains the histograms of observed particle packing fractions for each of the two algorithms.

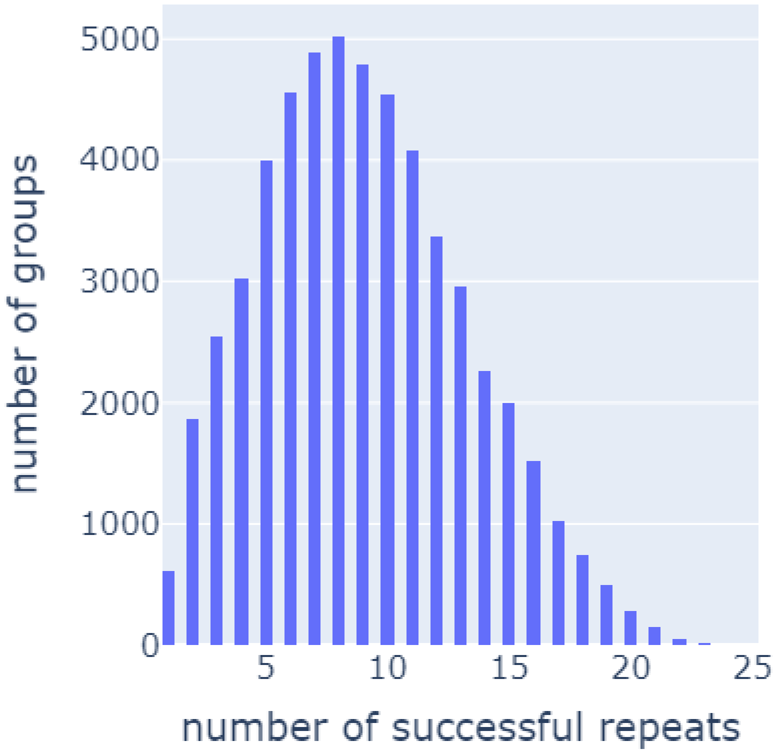

Figure 1. Histogram of number of parameter groups vs. number of successful repeats within a given group. For example, of the 65536 unique parameter combinations, approximately 5000 had eight successful repeats.

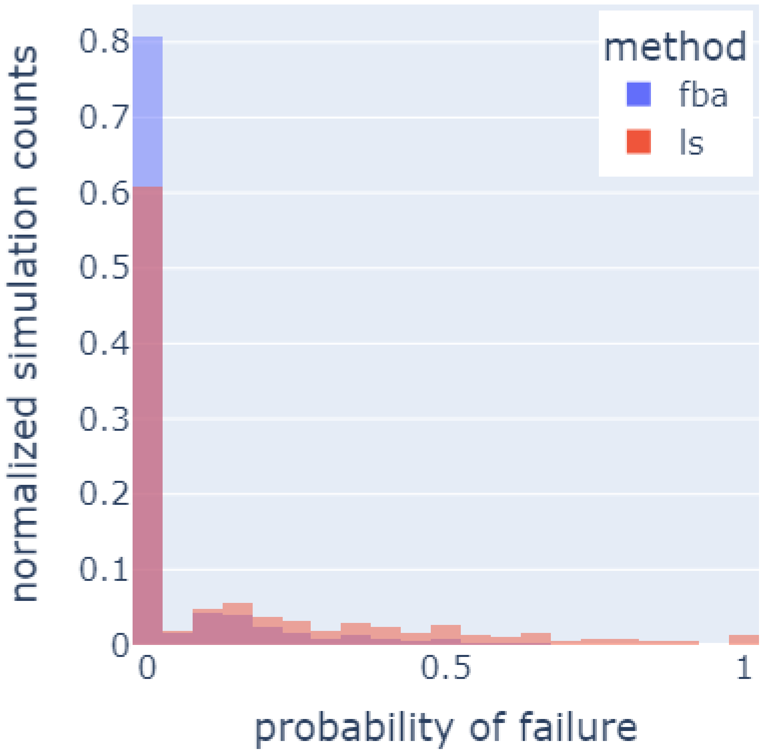

Figure 2. Histogram of normalized simulation counts vs. the probability of a simulation failing for a given parameter set. On average, the force-biased algorithm or fba (blue) is more likely to succeed than the Lubachevsky–Stillinger or ls (red) algorithm.

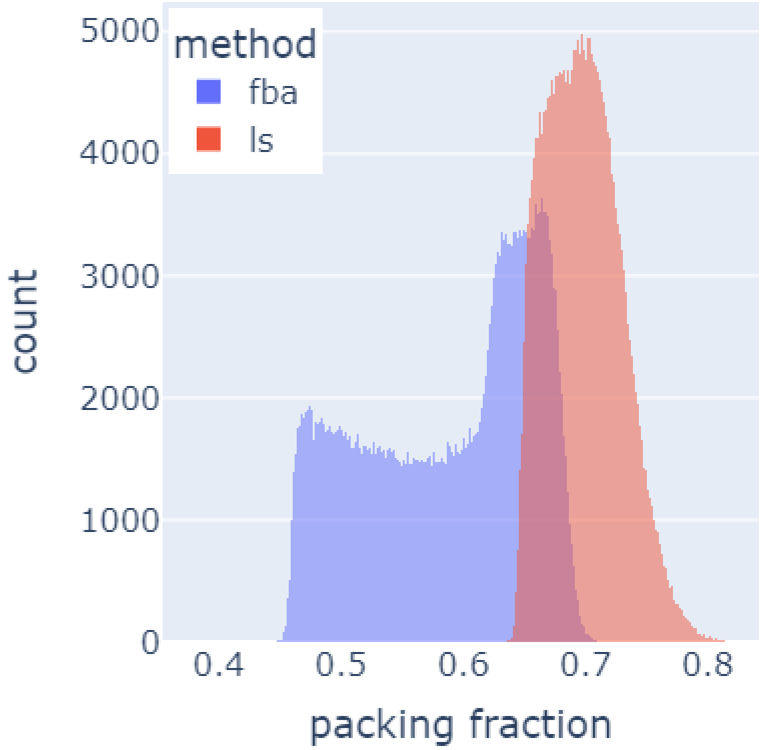

Figure 3. Histogram of number of simulations vs. packing fraction for the force-biased algorithm or fba (blue) and Lubachevsky–Stillinger or ls algorithm (red). On average, the ls algorithm tends to have higher packing fractions with a more Gaussian-like distribution than fba.

Experimental design, materials and methods

For this dataset, we aim to achieve a "Turing test" scenario where the surrogate model for a simulation is practically indistinguishable from the corresponding ground truth simulation. Here, we use https://github.com/VasiliBaranov/packing-generation[20] to run hard-sphere particle packing simulations while varying the particle size distribution. We ran repeat simulations to better capture noise, and we also tracked when simulations fail and the computational runtime at each step. Particle packing simulations were performed in a two-step process of a force-biased algorithm[15,16] followed by the Lubachevsky–Stillinger algorithm[17–19]. An attempt to run the LS algorithm was always preceded by an attempt to run the FBA algorithm. If the force-biased algorithm failed, the Lubachevsky–Stillinger algorithm was still attempted (https://github.com/sparks-baird/matsci-opt-benchmarks/blob/v0.2.2/src/matsci_opt_benchmarks/particle_packing/utils/packing_generation.py#L63-L183). The simulations were performed using mixtures of three different particle types, each characterized by two log-normal distribution parameters and three composition parameters. Two parameters (scale and shape) describe each of the three distributions, and three additional composition parameters describe the fractional share (e.g., in terms of volume) of each of the particle types. These nine parameters fully define the particle size distribution. With appropriate constraints applied, only seven of these parameters are necessary to fully define the particle size distribution. Additionally, the number of particles and an initial scaling factor were allowed to vary. With a greater number of particles, denser and more realistic packs can be generated at the expense of computational cost (i.e., the fidelity parameter). The initial scaling factor affects the computational stability of the simulation; with an adequate scaling factor, the simulation is more likely to be completed successfully. The quasi-random Sobol sampling technique was employed to generate parameter combinations, enabling a more uniform sampling of the allowable parameter space. We sampled 65536 unique parameter combinations. Repeat simulations for the parameter combinations were run to capture heteroskedastic noise, totaling 494498 simulations. To increase throughput and reduce latency, simulation parameters (including repeats) were shuffled and divided into batches, which were then dispatched to a high-performance computing environment for asynchronous evaluation. The data were recorded in a free-tier MongoDB Atlas database and then consolidated and prepared as datasets suitable for machine learning applications. Although it may serve other purposes, this dataset was primarily designed as a multi-fidelity benchmark dataset for constrained adaptive design experiments, hence the tracking of repeats, running simulations at various fidelities, incorporation of constraints, and tracking when simulations fail and the computational expense (whether or not the simulation runs successfully). For further implementation details, see https://github.com/sparks-baird/matsci-opt-benchmarks/tree/v0.2.2/scripts/particle_packing and https://github.com/sparks-baird/matsci-opt-benchmarks/tree/v0.2.2/notebooks/particle_packing. Instructions for model usage are available at https://matsci-opt-benchmarks.readthedocs.io/.

Ethics statements

There are no statements to declare.

CRediT author statement

Sterling G. Baird: Project administration, Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization. **Ramsey Issa**: Methodology, Software, Validation, Formal Analysis, Writing - Review & Editing. **Taylor D. Sparks**: Supervision, Funding acquisition

Acknowledgments

Funding: This work was supported by the National Science Foundation Division of Materials Research [Grant no. DMR-1651668].

We acknowledge Vasili Baranau and Robin De Schepper for help with the packing-generation codebase. We acknowledge Ramsey Issa and Hasan M. Sayeed for reviewing the manuscript and providing valuable feedback. We acknowledge Jason R. Hall for discussion of particle packing simulations in the context of industry processes. We acknowledge the University of Utah’s Center for High Performance Computing (CHPC) for providing computational resources.

We acknowledge OpenAI for providing free usage of their research tool, ChatGPT, which was used during the writing, review, and editing process.

Declaration of interests

x The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

References

(1) Baird, S. G. Materials Science Optimization Benchmark Dataset for Multi-Fidelity Hard-Sphere Packing Simulations, 2023. https://doi.org/10.5281/zenodo.7513019.

(2) Ghoreishi, S. F.; Molkeri, A.; Arróyave, R.; Allaire, D.; Srivastava, A. Efficient Use of Multiple Information Sources in Material Design. *Acta Materialia* **2019**, *180*, 260–271. https://doi.org/10.1016/j.actamat.2019.09.009.

(3) Kandasamy, K.; Vysyaraju, K. R.; Neiswanger, W.; Paria, B.; Collins, C. R.; Schneider, J.; Poczos, B.; Xing, E. P. Tuning Hyperparameters without Grad Students: Scalable and Robust Bayesian Optimisation with Dragonfly. *arXiv:1903.06694 [cs, stat]* **2020**.

(4) Hanaoka, K. Comparison of Conceptually Different Multi-Objective Bayesian Optimization Methods for Material Design Problems. *Materials Today Communications* **2022**, 103440. https://doi.org/10.1016/j.mtcomm.2022.103440.

(5) Häse, F.; Roch, L. M.; Aspuru-Guzik, A. Chimera: Enabling Hierarchy Based Multi-Objective Optimization for Self-Driving Laboratories. *Chem. Sci.* **2018**, *9* (39), 7642–7655. https://doi.org/10.1039/C8SC02239A.

(6) Baird, S. G.; Liu, M.; Sparks, T. D. High-Dimensional Bayesian Optimization of 23 Hyperparameters over 100 Iterations for an Attention-Based Network to Predict Materials Property: A Case Study on CrabNet Using Ax Platform and SAASBO. *Computational Materials Science* **2022**, *211*, 111505. https://doi.org/10.1016/j.commatsci.2022.111505.

(7) Eriksson, D.; Jankowiak, M. High-Dimensional Bayesian Optimization with Sparse Axis-Aligned Subspaces. *arXiv:2103.00349 [cs, stat]* **2021**.

(8) Baird, S.; Hall, J. R.; Sparks, T. D. The Most Compact Search Space Is Not Always the Most Efficient: A Case Study on Maximizing Solid Rocket Fuel Packing Fraction via Constrained Bayesian Optimization. ChemRxiv September 6, 2022. https://doi.org/10.26434/chemrxiv-2022-nz2w8-v2.

(9) Dunn, A.; Wang, Q.; Ganose, A.; Dopp, D.; Jain, A. Benchmarking Materials Property Prediction Methods: The Matbench Test Set and Automatminer Reference Algorithm. *npj Comput Mater* **2020**, *6* (1), 138. https://doi.org/10.1038/s41524-020-00406-3.

(10) De Breuck, P.-P.; Evans, M. L.; Rignanese, G.-M. Robust Model Benchmarking and Bias-Imbalance in Data-Driven Materials Science: A Case Study on MODNet. *J. Phys.: Condens. Matter* **2021**, *33* (40), 404002. https://doi.org/10.1088/1361-648X/ac1280.

(11) Wang, A.; Liang, H.; McDannald, A.; Takeuchi, I.; Kusne, A. G. Benchmarking Active Learning Strategies for Materials Optimization and Discovery. arXiv April 12, 2022. http://arxiv.org/abs/2204.05838 (accessed 2022-07-04).

(12) Liang, Q.; Gongora, A. E.; Ren, Z.; Tiihonen, A.; Liu, Z.; Sun, S.; Deneault, J. R.; Bash, D.; Mekki-Berrada, F.; Khan, S. A.; Hippalgaonkar, K.; Maruyama, B.; Brown, K. A.; Fisher III, J.; Buonassisi, T. Benchmarking the Performance of Bayesian Optimization across

Multiple Experimental Materials Science Domains. *npj Comput Mater* **2021**, *7* (1), 188. https://doi.org/10.1038/s41524-021-00656-9.

(13) Henderson, A. N.; Kauwe, S. K.; Sparks, T. D. Benchmark Datasets Incorporating Diverse Tasks, Sample Sizes, Material Systems, and Data Heterogeneity for Materials Informatics. *Data in Brief* **2021**, *37*, 107262. https://doi.org/10.1016/j.dib.2021.107262.

(14) Häse, F.; Aldeghi, M.; Hickman, R. J.; Roch, L. M.; Christensen, M.; Liles, E.; Hein, J. E.; Aspuru-Guzik, A. Olympus: A Benchmarking Framework for Noisy Optimization and Experiment Planning. *Mach. Learn.: Sci. Technol.* **2021**, *2* (3), 035021. https://doi.org/10.1088/2632-2153/abedc8.

(15) Mościński, J.; Bargieł, M.; Rycerz, Z. A.; Jacobs, P. W. M. The Force-Biased Algorithm for the Irregular Close Packing of Equal Hard Spheres. *Molecular Simulation* **1989**, *3* (4), 201–212. https://doi.org/10.1080/08927028908031373.

(16) Bezrukov, A.; Bargieł, M.; Stoyan, D. Statistical Analysis of Simulated Random Packings of Spheres. *Particle & Particle Systems Characterization* **2002**, *19* (2), 111–118. https://doi.org/10.1002/1521-4117(200205)19:2<111::AID-PPSC111>3.0.CO;2-M.

(17) Skoge, M.; Donev, A.; Stillinger, F. H.; Torquato, S. Packing Hyperspheres in High-Dimensional Euclidean Spaces. *Phys. Rev. E* **2006**, *74* (4), 041127. https://doi.org/10.1103/PhysRevE.74.041127.

(18) Lubachevsky, B. D. How to Simulate Billiards and Similar Systems. *Journal of Computational Physics* **1991**, *94* (2), 255–283. https://doi.org/10.1016/0021-9991(91)90222-7.

(19) Lubachevsky, B. D.; Stillinger, F. H. Geometric Properties of Random Disk Packings. *Journal of Statistical Physics* **1990**, *60* (5), 561–583. https://doi.org/10.1007/BF01025983.

(20) Baranau, V.; Tallarek, U. Random-Close Packing Limits for Monodisperse and Polydisperse Hard Spheres. *Soft Matter* **2014**, *10* (21), 3826–3841. https://doi.org/10.1039/C3SM52959B.