Open Access

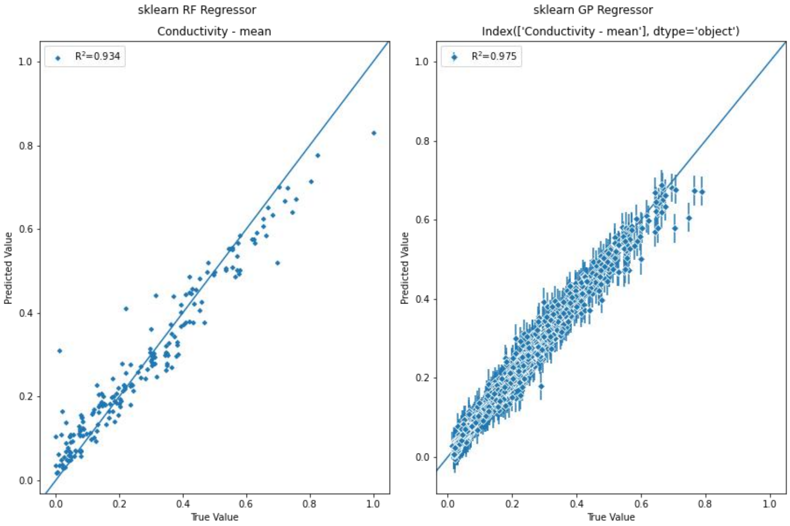
Supplementary Material

Mapping pareto fronts for efficient multi-objective materials discovery

Implementation of real-world problems

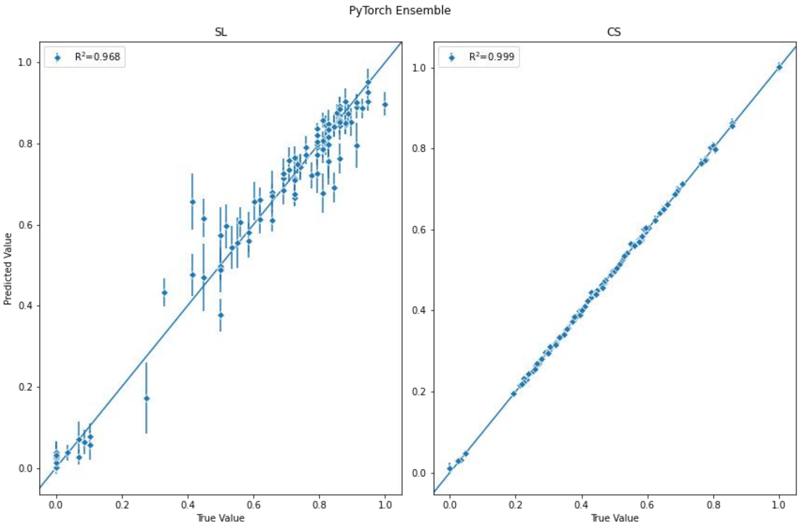
The datasets are first extracted from the relevant papers, and undergo a preliminary scoring for various standard regressors from scikit-learn[1] with an 80/20 train-test split to determine the best model with the highest R2 score and lowest MSE. Afterwards, the best model is then selected by retraining on 100% of the data.

For thin film[2], RandomForestRegressor was selected. However, due to the stepped outputs for decision tree-based regressors, we opted to perform a 2-level training. Since RF has been previously shown to provide good interpolation ability for predictions, we assume that it models the underlying dataset well. We took 1000 virtually generated samples through Latin Hypercube sampling, which is then fed into a GaussianProcessRegressor to provide a more continuous response surface that can be used to benchmark optimization.



Supplementary Figure 1. Scatter plot of actual versus predicted outputs for the models used in thin film. There is only 1 modelled input (conductivity) since the other optimization objective is the input variable for process temperature.

As for concrete slump[3], we opted to go with an ensemble of PyTorch[4] deep neural networks instead due to better accuracy. For each objective, 10 separate neural networks with a similar architecture are trained, and the mean and standard deviation of their predictions are taken as the output. We omitted the 3rd objective for flow due to it being similar to slump.



Supplementary Figure 2. Scatter plot of actual versus predicted outputs for the models used in concrete slump.

Constraint handling for real-world problems

As compared to optimization of synthetic benchmarks, feasibility is critical for real-world experimentation since it is physically impossible to validate an unfeasible material or process. As such, every single candidate proposed during the optimization must meet a feasibility constraint. This can be done by ‘repairing’ the inputs. For example, in combinatorial screening where parameters must sum to 100% due to Molar/weight balances, we can implement a repair operator that enforces:

$$\Sigma \boldsymbol{x}' = \Sigma \boldsymbol{x} \,/\, (\Sigma \boldsymbol{x}/100)$$

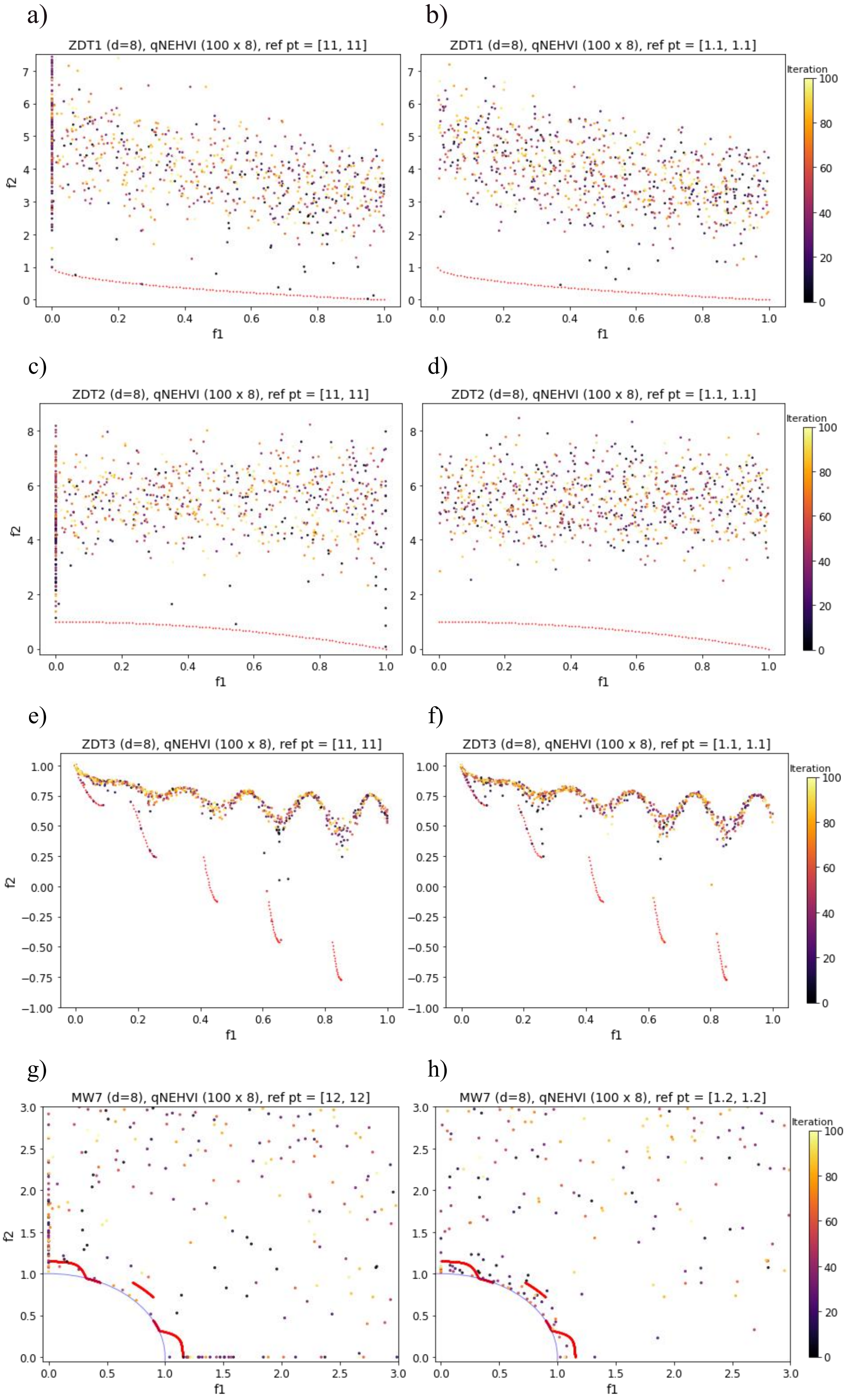
Alternatively, such rules can also be transformed into an inequality constraint following the tutorial on pymoo website[5], for example:

Taking the base equality form $g(\boldsymbol{x})$: $\Sigma \boldsymbol{x} - 100 = 0$

Converting into inequality form $g(\boldsymbol{x})$: $(\boldsymbol{x} - 100)^2 - \epsilon \leq 0$

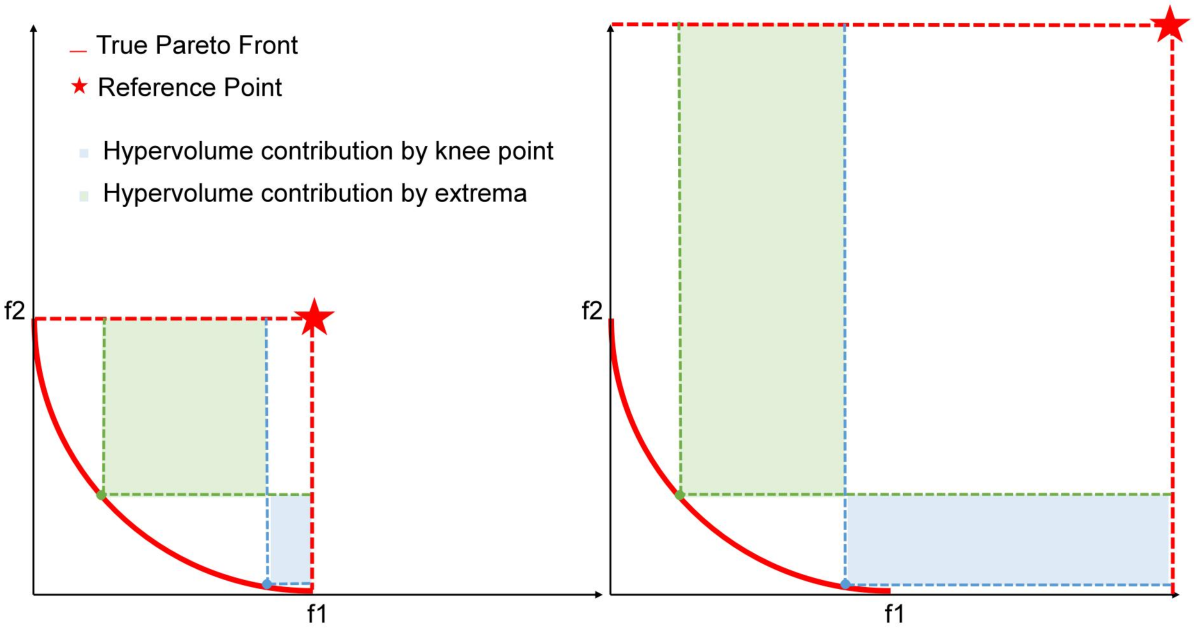
A good rule of thumb is to set ϵ as 0.1% of the base value to allow for some margin in rounding off errors. The setting of constraints helps ensure that the optimization is actively achieving feasibility first, thereby minimizing the need to repair inputs. This becomes particularly important for surrogate-based methods like qNEHVI, as integrating constraints into the optimization mechanism is shown to have advantages in producing feasible solutions[6].

Selecting reference point for hypervolume-based Bayesian optimization



Supplementary Figure 3. Optimization trajectory in objective space for a single optimization run of 100 iterations × 8 points per batch. (a-b) ZDT1. (c-d) ZDT2. (e-f) ZDT3. (g-h) MW7. The reference point is changed from large to small, keeping all other hyperparameters constant. The red line represents the true PF, while MW7 being a constrained problem has an additional blue line to show the unconstrained PF. The color of each experiment refers to the number of iterations. Results here indicate a bias to extrema for large reference points, which is consistent with literature.

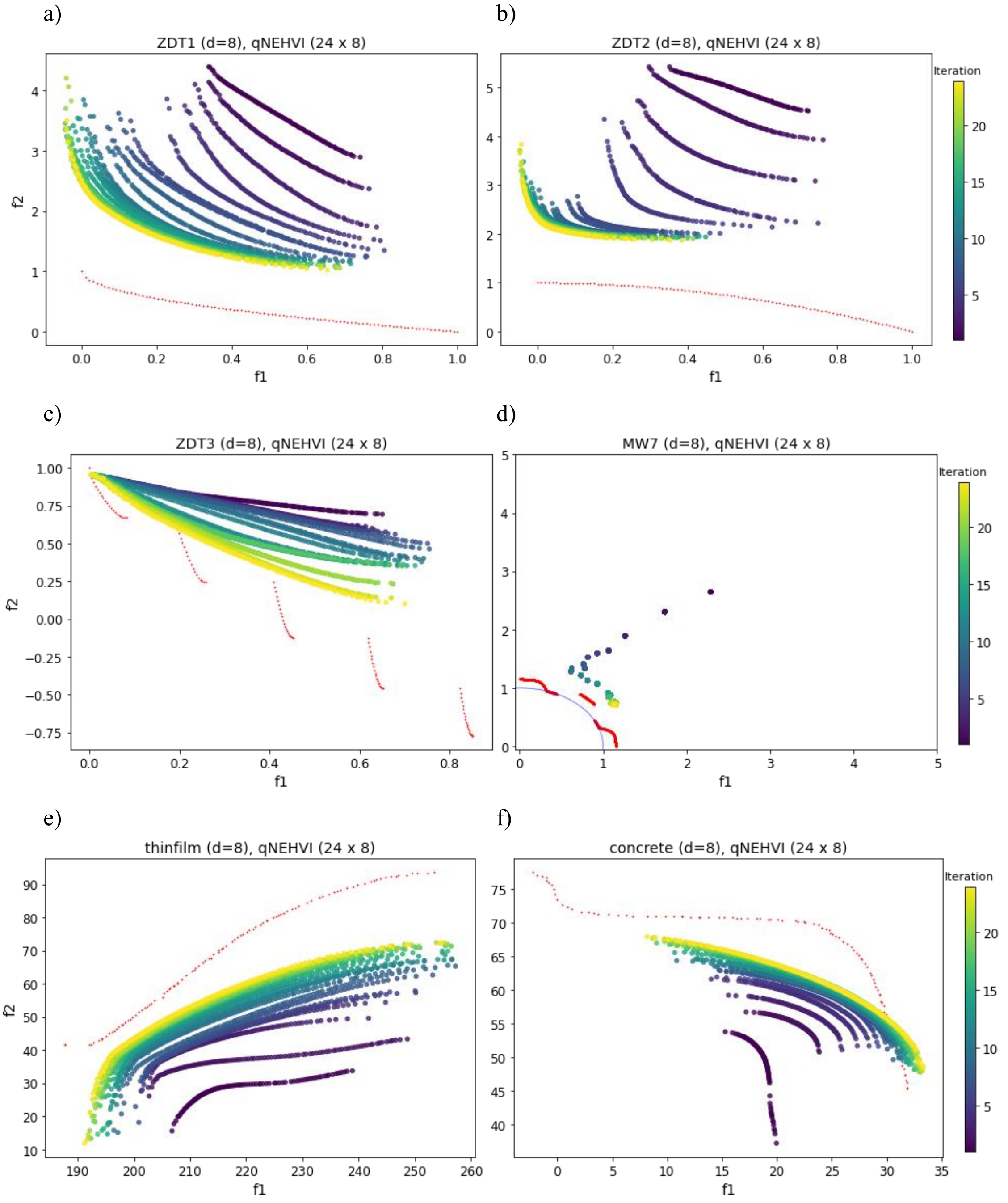
For minimization problems in particular, the choice of reference point is crucial to the performance of hypervolume-based optimization strategies[7]. Ishibushi *et al.*[8] recommend for minimization problems to set the reference point to the nadir - 0.1 * range where 0.1 provides a robust scaling factor for most problems. Empirically, it is found that a large reference point (towards infinity) biases extrema, and conversely, a small reference point (close to PF) values knee points[9,10]. We corroborate their findings here and show for ZDT1, ZDT2 and MW7 in Figures 1a, 1c and 1g that the larger reference point does cause qNEHVI to propose many points at the extrema of $f_1 = x_1$, as well as at f_2 for MW7, while the smaller reference point in 1b, 1d and 1h provides better distribution of solutions. We hypothesize that with a larger reference point that covers more of the objective space, any points which can 'cover' more space (i.e., the extrema) provide greater HV improvement contribution. In contrast, a smaller reference point near the PF would not have this problem, where knee points give greater cover. We illustrate in Figure 2 our explanations for this.


True Pareto Front
Reference Point
Hypervolume contribution by knee point
Hypervolume contribution by extrema
f2
f1
f2
f1

Supplementary Figure 4. Illustration of different reference points in bi-objective space for a convex minimization problem. Note the ratios of HV covered by the knee point in green versus the extreme point in blue. As the reference point increases, the relative contribution of extrema increases in comparison to that of the knee point.

Notably, for problem ZDT2, setting a small reference point, as shown in Figure 1d, could totally prevent qNEHVI from reaching the PF. As discussed in the main text, qNEHVI relies heavily on QMC sampling to generate candidates that are then evaluated for their expected HV improvement. If the bulk of candidates is generated above the reference point, this effectively means that no points are able to provide improvement, and qNEHVI thus fails to exploit the PF for the GP surrogate model to learn more.

Advancement of Pareto Front knowledge in Gaussian surrogate models

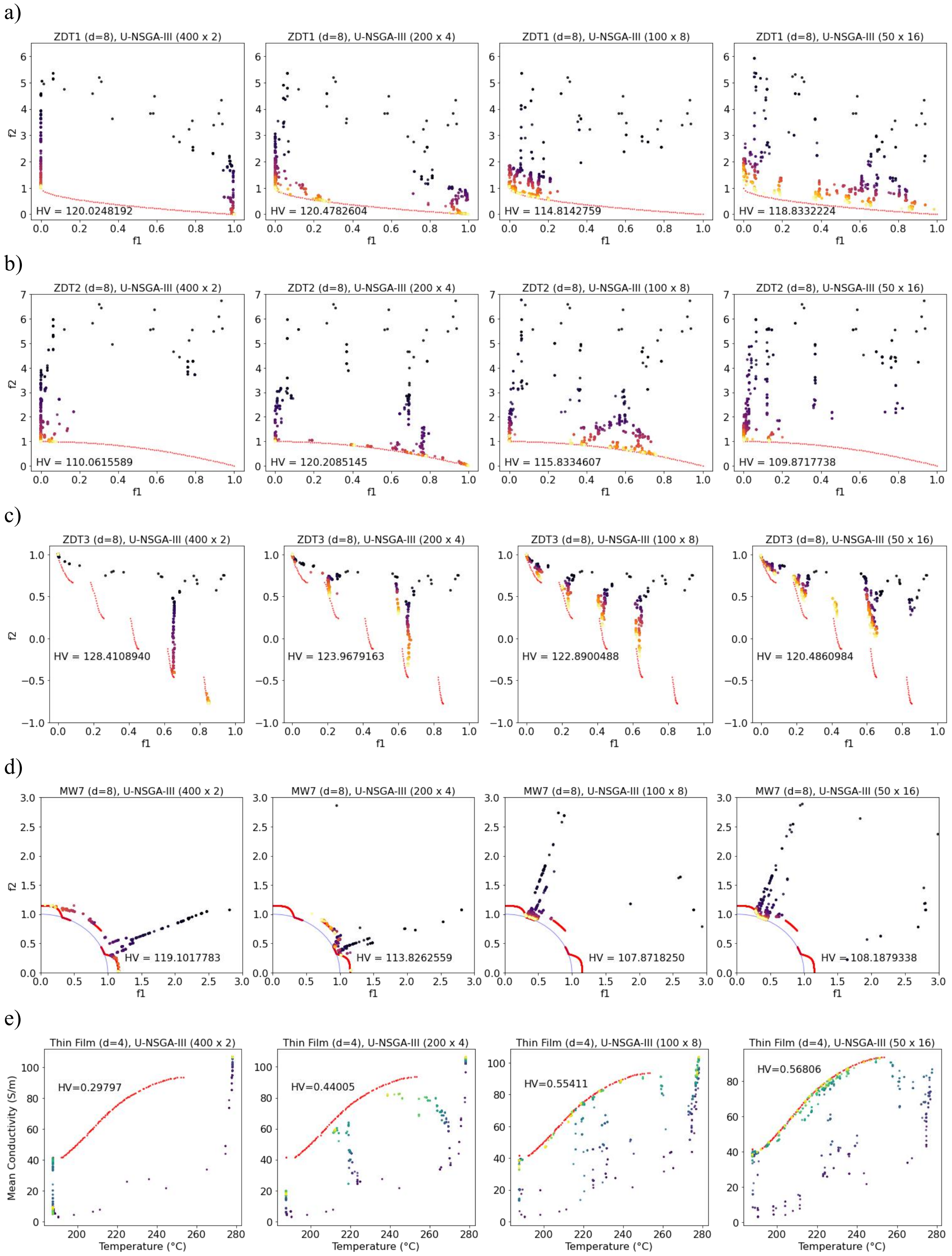


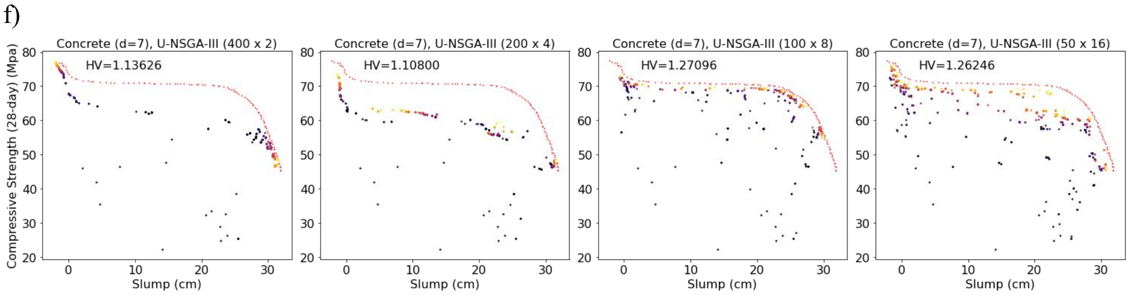
Supplementary Figure 5. Pareto Front of GP surrogate model in objective space at each iteration for a single optimization run of 24 iterations x 8 points per batch. (a) ZDT1. (b) ZDT2. (c) ZDT3. (d) MW7. (e) Thin Film. (f) Concrete Slump. The updated GP surrogate model in qNEHVI at each iteration is separately solved with U-NSGA-III, taking a population size of 100 and the number of generations at 100, and the final population is plotted in objective space to represent the model's known PF. We limited ourselves to the smaller evaluation budget due to long run times for solving the surrogate model every iteration. Results here indicate that within a limited evaluation budget of 192 points (24 iterations × 8 points per batch), the underlying GP surrogate model fails to clearly model the true PF in red.

We perform a single optimization run for 24 iterations × 8 points per batch on all the synthetic and real-world benchmarks for qNEHVI, taking the learned GP surrogate model at each iteration and finding its solved PF, like the approach taken by TSEMO[11]. We find that in all cases, the underlying surrogate model completely fails to properly model the true PF, which indicates that the gain in HV due to the qNEHVI acquisition function is being mainly driven by the stochasticity of QMC sampling in providing candidates, rather than clear knowledge of the function and objective space.

Notably, we also observe in Figure 3f for the Concrete Slump problem that the identified PF by the GP model crosses the true PF at a specific region, which we observed is where the highest probability of points is concentrated for the main text in Figure 8d. This is consistent with our conclusion for the other benchmarks and problems, since the GP model is identifying mostly a false PF, with certain regions being correct.

Optimization trajectories at different batch sizes for U-NSGA-III





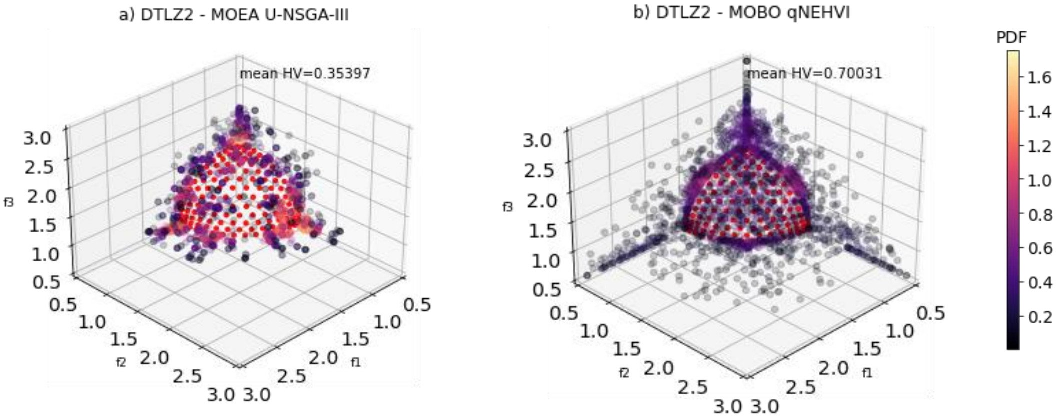
Supplementary Figure 6. Optimization trajectory across objective space for different population sizes in U-NSGA-III. a) ZDT1. b) ZDT2. c) ZDT3. d) MW7. e) Thin Film. f) Concrete Slump. We keep the same total evaluation budget of 800 points, excluding initialization. Color bar was omitted to conserve space.

We also plot out the optimization trajectory for U-NSGA-III to better illustrate convergence for different populations. While larger populations have better coverage, this is often at the expense of total iterations to allow for the exploitation of the entire PF. This can be clearly illustrated by lower population sizes having enough generations to exploit the near-optimal region, as seen by the trajectory of solutions for smaller batch sizes in Figures 4a, 5b and 5d for ZDT1, ZDT2 and MW7, respectively.

Notably, we observe in Figure 4c for ZDT3 that a smaller population fails to maintain solutions across the disconnected space, since there are only limited solutions that can drive evolution. This means that entire subregions of objective space are completely missed out, and is a clear limitation of population-based MOEAs which require maintaining the diversity of solutions[12,13].

We also note in Figure 4e and 4f that larger population sizes are required to converge towards PF for real-world problems. We hypothesize that the more mathematically complex real-world problems contain various local optima in which members of the population can be 'trapped' in, and are unable to evolve towards the true PF effectively. It is unclear as to which population size is ideal for a specific problem, although there is various literature that explores this in closer detail[14,15].

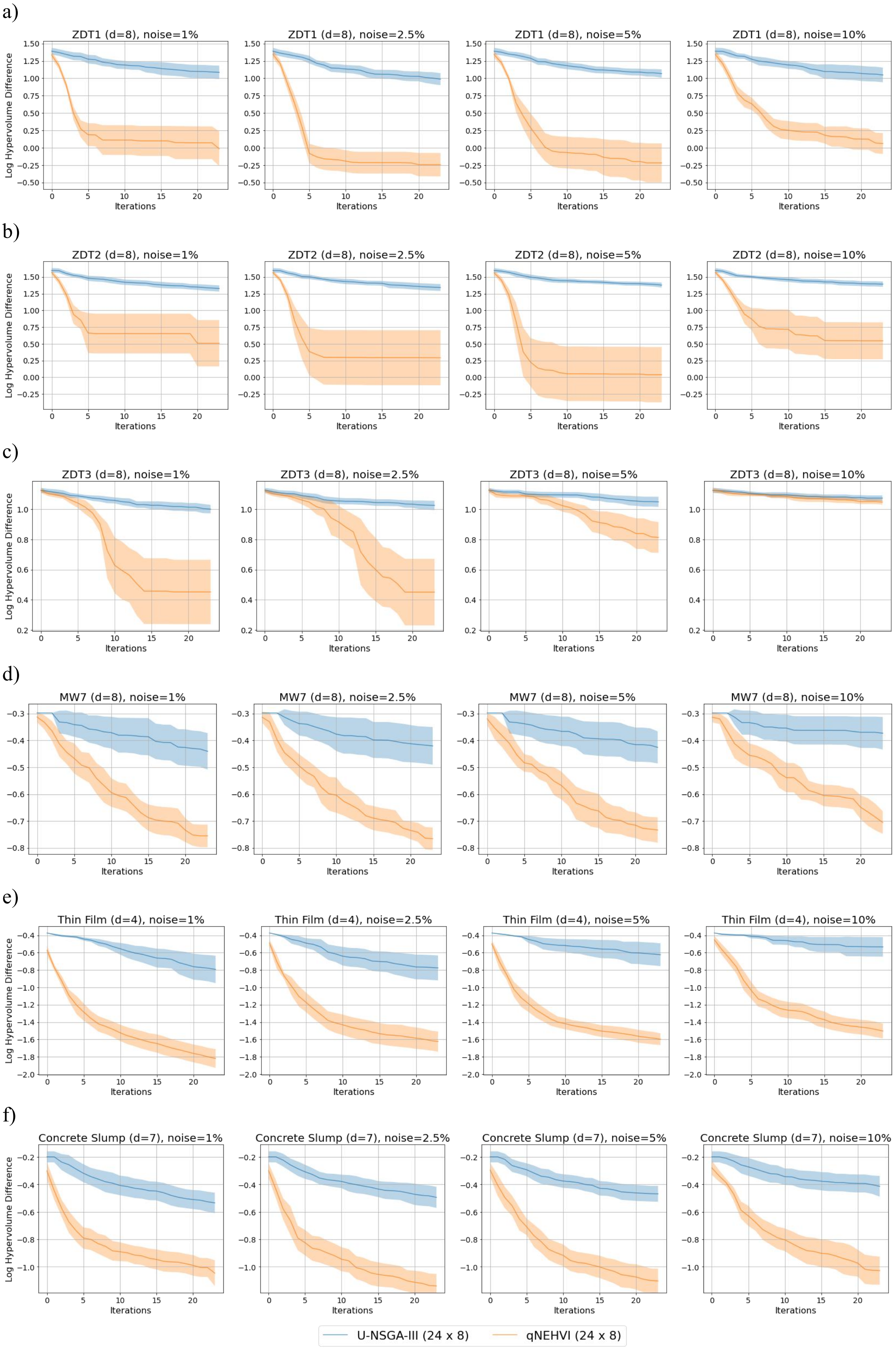
3-objective Optimization



Supplementary Figure 7. Probability density maps for both qNEHVI and U-NSGA-III on 3-objective 8-variable DTLZ2 problem with 24 iterations x 8 evaluations per batch.

We also performed optimization for a 3-objective problem to showcase that the metrics can be utilized, although it is more difficult to analyze in 3D space. We recommend that users should instead plot these onto orthogonal projections, looking at only 2 dimensions at a given time, or instead use the inline plotting function in Jupyter to create interactive plots that can be rotated accordingly. The results shown here are consistent with our conclusions regarding qNEHVI vs U-NSGA-III in the main text.

Log hypervolume difference at different output noise levels



Supplementary Figure 8. Log hypervolume difference versus iterations for both algorithms at different noise levels. a) ZDT1. b) ZDT2. c) ZDT3. d) MW7. e) Thin Film. f) Concrete Slump. Results indicate robust performance from both algorithms up to 10%, with sharp drop in ZDT3.

Following the same restricted budget of 24 iterations x 8 per batch, we then compared amounts of white noise added to objectives. We theorize that noise in observations does not significantly affect the performance MOEAs, since the generation of new samples is done stochastically via crossover and mutation, especially for larger population sizes. Conversely, noise has been directly accounted for within qNEHVI as part of the probability distribution and uncertainty in sampling, which leads to robust performance. We note in Figure 8c that the performance of both algorithms falls off sharply for ZDT3 with added noise, which can be attributed to the very disjointed PF being sensitive to noise. This indicates that overly small feasible regions at the PF are likely to be negatively affected by noise, which can easily shift the sample out of feasibility.

Supplementary Table 1. Wall time results in seconds

	MOEA U-NSGA-III				**MOBO qNEHVI**		
	96 by 2	48 by 4	24 by 8	12 by 8	96 by 2	48 by 4	24 by 8
ZDT1	0.75	**0.73**	0.78	0.81	277.19	196.85	**191.53**
	+/-0.03	**+/-0.05**	+/-0.04	+/-0.12	+/-11.95	+/-6.67	**+/-6.13**
ZDT2	0.76	**0.72**	0.79	0.81	264.25	195.91	**180.61**
	+/-0.06	**+/-0.07**	+/-0.06	+/-0.10	+/-16.19	+/-9.41	**+/-8.89**
ZDT3	0.77	**0.75**	0.81	0.87	339.13	**224.29**	270.71
	+/-0.05	**+/-0.02**	+/-0.06	+/-0.12	+/-17.57	**+/-14.93**	+/-11.3
MW7[1]	1.55	**1.04**	1.13	1.13	576.07	542.58	**511.44**
	+/-1.06	**+/-0.02**	+/-0.03	+/-0.10	+/-52.88	+/-67.98	**+/-31.91**

REFERENCES

1. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine learning in Python. the Journal of machine Learning research. 2011;12:2825-30. Available from: https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf?ref=https:/ [Last accessed on 7 Mar 2023]
2. MacLeod BP, Parlane FGL, Rupnow CC, et al. A self-driving laboratory advances the Pareto front for material properties. *Nat Commun* 2022;13:1-10. [DOI: 10.1038/s41467-022-28580-6]
3. Yeh IC. Modeling slump of concrete with fly ash and superplasticizer. *Comput Concr An Int J* 2008;5:559-572. Available from: https://www.dbpia.co.kr/Journal/articleDetail?nodeId=NODE10903242 [Last accessed on 7 Mar 2023]
4. Paszke A, Gross S, Massa F, et al. Pytorch: An imperative style, high-performance deep learning library. Adv Neural Inf Process Syst. 2019;32. Available from: https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf [Last accessed on 7 Mar 2023]
5. Blank J, Deb K. Pymoo: Multi-objective optimization in python. IEEE Access. 2020;8:89497-89509. [DOI: 10.1109/ACCESS.2020.2990567]

[1] Only 6 runs attained.

6. Gardner JR, Kusner MJ, Xu ZE, Weinberger KQ, Cunningham JP. Bayesian optimization with inequality constraints. Available from: http://proceedings.mlr.press/v32/gardner14.pdf [Last accessed on 7 Mar 2023]

7. Auger A, Bader J, Brockhoff D, Zitzler E. Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications. *Theor Comput Sci* 2012;425:75-103.

8. Ishibuchi H, Akedo N, Nojima Y. A many-objective test problem for visually examining diversity maintenance behavior in a decision space. [DOI:10.1145/2001576.2001666]

9. Jiang S, Yang S, Li M. On the use of hypervolume for diversity measurement of Pareto front approximations. In: 2016 IEEE Symposium Series on Computational Intelligence (SSCI); 2016:1-8. [DOI:10.1109/SSCI.2016.7850225]

10. Yang K, Emmerich M, Deutz A, Bäck T. Multi-objective Bayesian global optimization using expected hypervolume improvement gradient. *Swarm Evol Comput* 2019;44:945-956.

11. Bradford E, Schweidtmann AM, Lapkin A. Efficient multiobjective optimization employing Gaussian processes, spectral sampling and a genetic algorithm. *J Glob Optim* 2018;71:407-38.

12. Li K, Chen R, Fu G, Yao X. Two-archive evolutionary algorithm for constrained multiobjective optimization. *IEEE Trans Evol Comput* 2018;23:303-315.

13. Pan L, He C, Tian Y, Su Y, Zhang X. A region division based diversity maintaining approach for many-objective optimization. *Integr Comput Aided Eng* 2017;24. [DOI:10.3233/ICA-170542]

14. Tanabe R, Oyama A. The impact of population size, number of children, and number of reference points on the performance of NSGA-III. In: International Conference on Evolutionary Multi-Criterion Optimization. Springer; 2017:606-21.

15. Hort M, Sarro F. The effect of offspring population size on NSGA-II: a preliminary study. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion; 2021:179-180.