Electronic Supplementary Material (ESI) for Journal of Materials Chemistry A.

Supplementary Information

Finely Tuned Inverse Design of Metal-Organic Frameworks with User-desired Xe/Kr Selectivity

*Yunsung Lim, Junkil Park, Sangwon Lee, Jihan Kim**

Department of Chemical and Biomolecular Engineering, Korea Advanced Institute of Science and Technology (KAIST), 291, Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea

Correspondence to: jihankim@kaist.ac.kr

CONTENTS

S1. Supplementary figures for the machine learning results and initial training set

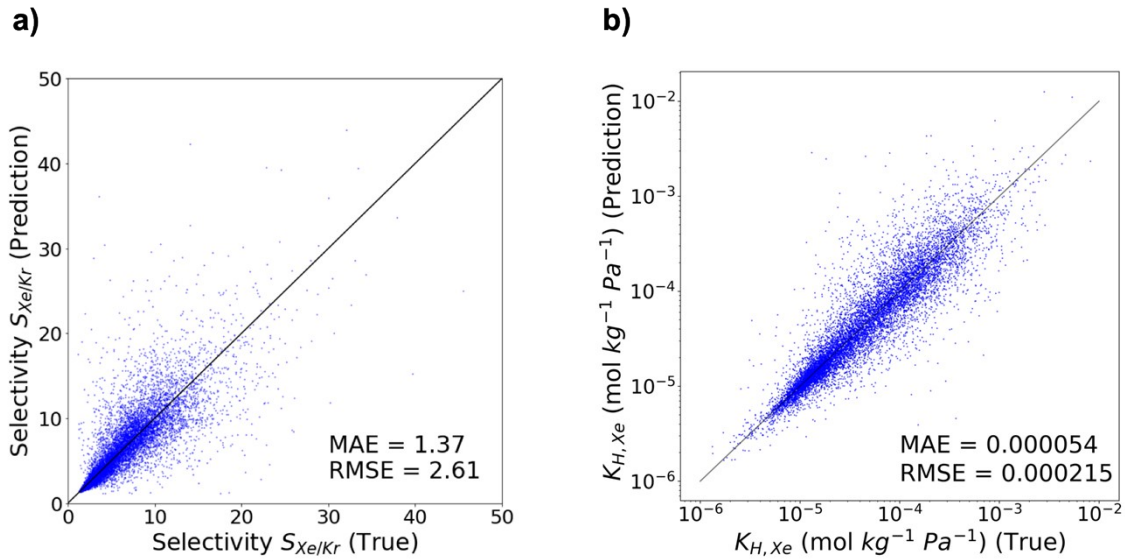

Fig. S1 Prediction results of the machine learning model, MOF-NET. The model is trained with frameworks after cycle 3. a) Predicted selectivity versus true selectivity. b) Predicted $K_{H,Xe}$ versus true $K_{H,Xe}$. The model showed high accuracy in low selectivity and low $K_{H,Xe}$ region which helped the genetic algorithm to sort out low performing structures well enough.

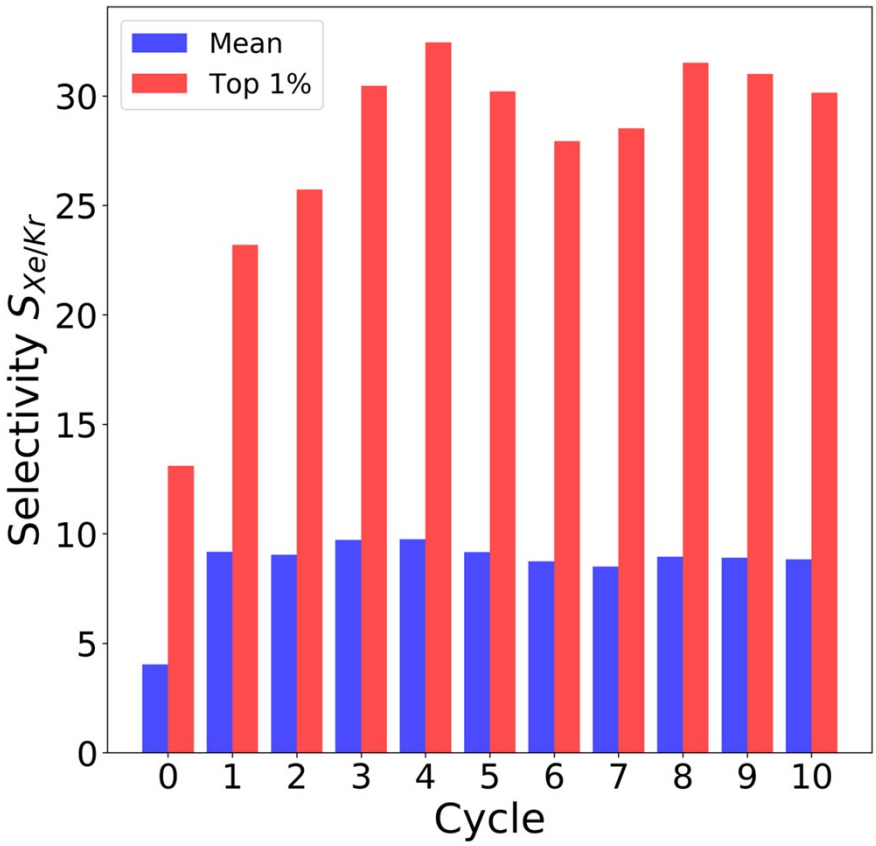

Fig. S2 Cycle versus selectivity. Blue bar denotes mean selectivity value of MOFs and red bar denotes selectivity value of top 1% MOF for each cycle, respectively. There were no dramatic changes after cycle 3 for both mean selectivity value and selectivity value of top 1% MOF.

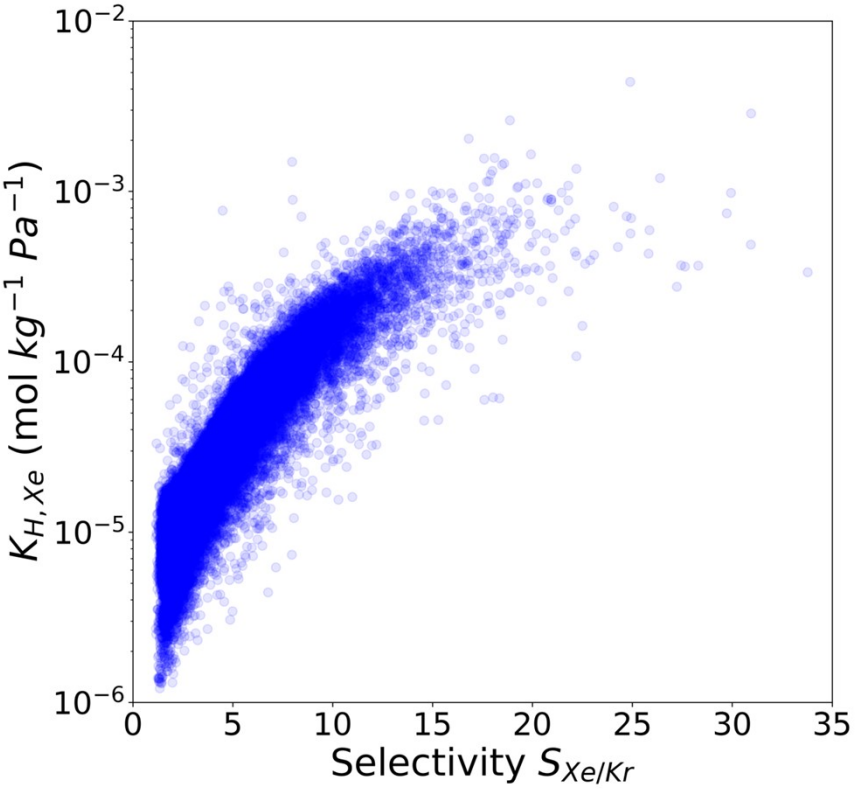

Fig. S3 $K_{H,Xe}$ versus selectivity plot for initial randomly generated frameworks. ($K_{H,Xe} < 10^{-6}$ mol kg^{-1} Pa^{-1} were eliminated). As shown in figure, $K_{H,Xe}$ gradually increases with selectivity.

S2. Additional information for MOF candidates with record-breaking Xe/Kr selectivity

To validate accuracy of simulations via in-house GPU code,[1] we simulated 16 promising candidates with both in-house GPU code and RASPA software package[2] for $K_{H,Xe}$. RASPA software package slightly overestimated $K_{H,Xe}$ compared to in-house GPU code, but every 16 frameworks have nearly similar values in both simulation results (**Fig. S4**). This discrepancy is attributed to the difference in software implementations. Unlike RASPA software package, our in-house GPU code is encouraged to use energy grid for most of cases. Thus, GPU code obtains potential energy from discrete points within energy grids via linear interpolation functions.

3 node building blocks (NBBs) were involved in 2 chosen MOF candidates, htp+N5+N270+E0 and htp+N5+N92+E0. N5, benxenehexacarboxylate moiety (BHC),[3] was used as organic linker for both frameworks. N92 (Indium metal cluster from REFCODE : IMUTUG_ion_b)[4] and N270 (Gallium metal cluster from REFCODE : HOMZEP_clean)[5] were respectively used as metal clusters for each candidate.

	htp+N5+N270+E0	**htp+N5+N92+E0**
Selectivity (rigid, GPU)	100	78
Selectivity (rigid, RASPA)	110	78
$K_{H,Xe}$ (mol kg^{-1} Pa^{-1}, GPU)	1.6×10^{-1}	5.8×10^{-2}
$K_{H,Xe}$ (mol kg^{-1} Pa^{-1}, RASPA)	1.7×10^{-1}	6.3×10^{-2}
Xe Q_{st} (kJ mol^{-1})	44	41
Kr Q_{st} (kJ mol^{-1})	31	29
Selectivity (flex)	97	75
Building blocks	Node : N5, N270 Edge : none	Node : N5, N92 Edge : none

Table S1 Details on properties ($S_{Xe/Kr}$ (rigid), $S_{Xe/Kr}$ (flex), $K_{H,Xe}$) and components of 2 promising candidates, htp+N5+N270+E0 and htp+N5+N92+E0.

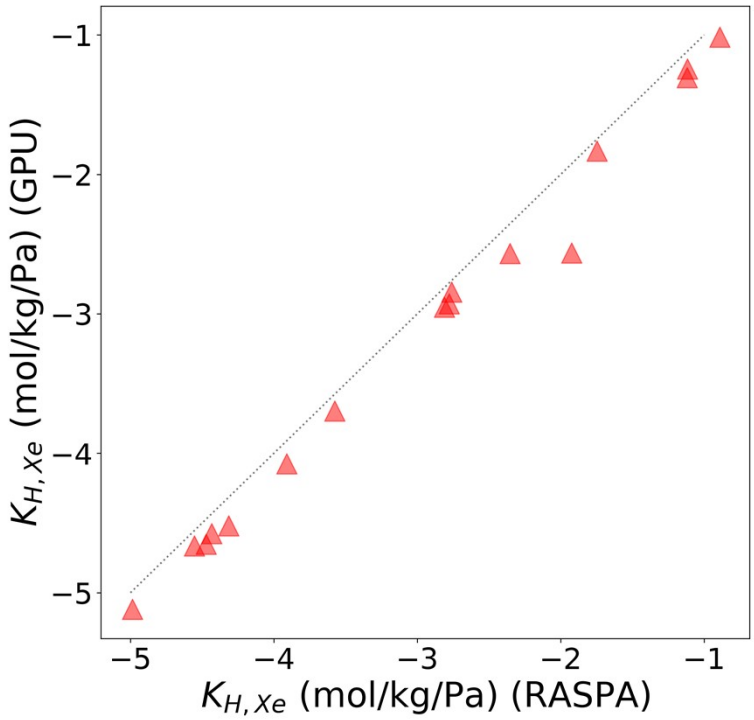

Fig. S4 RASPA versus in-house GPU code for the MC simulations of $K_{H,Xe}$ without blocking algorithm. The axis was converted to log scale for convenience.

Framework	**Normalized Energy (Energy / # of metal)**	**Energy (kcal/mol)**
nia+N5+N270+E0	200	1200
std+N5+N270+N5+E0+E0	210	2500
htp+N5+N270+E0	**260**	**3100**
ste+N5+N270+N270+E0+E0	280	3300
std+N270+N5+N270+E0+E0	310	3700
nia+N270+N5+E0	350	2100
sss+N270+N5+N270+E0+E0	370	3300
ste+N270+N5+N5+E0+E0	390	4700
htp+N270+N5+E0	400	4800
sss+N5+N270+N5+E0+E0	440	4000

Table S2 Energy calculation results for polymorphic structure of htp+N5+N270+E0

Framework	**Normalized Energy (Energy / # of metal)**	**Energy (kcal/mol)**
nia+N5+N92+E0	170	1000
std+N5+N92+N5+E0+E0	180	2100
htp+N5+N92+E0	**230**	**2700**
ste+N5+N92+N92+E0+E0	240	2900
std+N92+N5+N92+E0+E0	260	3100
nia+N92+N5+E0	310	1800
sss+N92+N5+N92+E0+E0	310	2800
ste+N92+N5+N5+E0+E0	340	4000
htp+N92+N5+E0	340	4100
sss+N5+N92+N5+E0+E0	370	3300

Table S3 Energy calculation results for polymorphic structures of htp+N5+N92+E0

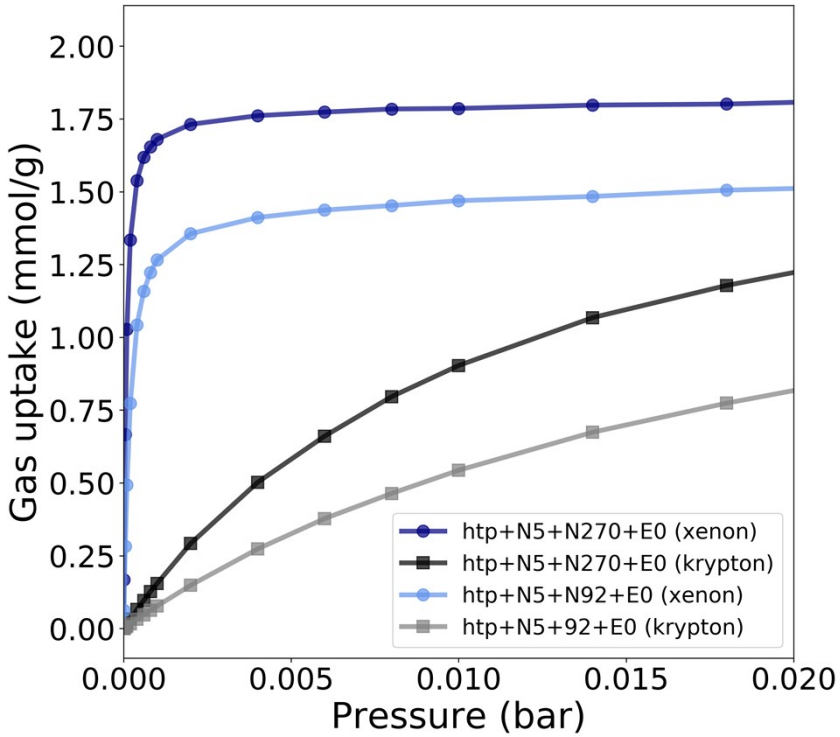

Fig. S5 Adsorption isotherm of htp+N5+N270+E0 and htp+N5+N92+E0 with xenon and krypton, respectively.

S3. Details on cost functions of genetic algorithm

We devised a new cost function for the genetic algorithm for inverse design of MOFs with various kinds of conditions. Conventional cost function to generate record-breaking MOFs was simply to predict selectivity value of frameworks by the machine learning model (**Fig. S6a**).[6] Crossover and mutation occured during the genetic algorithm and frameworks were optimized to have high selectivity value.

S3-1. Cost function for one user-desired property

In this case, cost function was changed to predict difference between the prediction value (obtained from output of the machine learning model) and targeted user-desired value of properties. For example, when one wants to discover MOFs with selectivity of 5, then a fitness value (result of cost function) is absolute value of predicted selectivity minus 5 (**Fig. S6b**). Then, the genetic algorithm scratches frameworks from the vast MOF spaces which have as small fitness value as possible to satisfy the user-desired selectivity of 5. Same approach was applied in $K_{H,Xe}$ also. Every $K_{H,Xe}$ was normalized with log scale to directly apply the weights of machine learning model which designed to predict the value of properties within 0 ~ 1.

S3-2. Cost function for two user-desired property simultaneously.

This case is much more complicate than the previous case. Generally, the scales of properties are different from each other. For example, Xe/Kr selectivity is range from 0 to 100 or more but $K_{H,Xe}$ is range from 10^{-1} to under 10^{-6} or less. To compare two different properties, we standardized the value of properties by the mean and standard deviation value of training set. We assumed that the distribution of properties may follow Gaussian distribution for large amounts of data for standardization. A fitness value is distance between standardized targeted value and standardized predicted value (**Fig. S6c**).

The algorithm gradually optimizes frameworks to reduce the gap between two points to achieve pareto optimization for both conditions.

S3-3. Cost function for minimum limitation of selectivity

Some people may conceive ramp function as a suitable cost function to generate frameworks which can exceed specific minimum threshold of selectivity. However, ramp function showed poor performance in this work because of the low prediction accuracy of the machine learning model in high selectivity regime. To overcome this problem, we used the idea of the cost function in section S3-1 to devise a new cost function. As aforementioned, it is possible to optimize frameworks to have specific value of selectivity. Thus, we manipulated cost function to follow ramp function which set the minimum limitation as a point of inflection and 30% improvement of the minimum limitation as an end point. Then, the fitness value gradually decreases for higher selectivity regime (**Fig. S6d**). For example, if one wants to find MOFs with selectivity higher than 5, a framework ($S_{Xe/Kr} = 6.5$) has the largest fitness value and another framework of selectivity ($S_{Xe/Kr} = 4$) has zero for fitness value, and the other framework ($S_{Xe/Kr} = 7$) has quite small fitness value. The algorithm optimizes frameworks to have large fitness value to which help higher selectivity value than minimum threshold value be obtained.

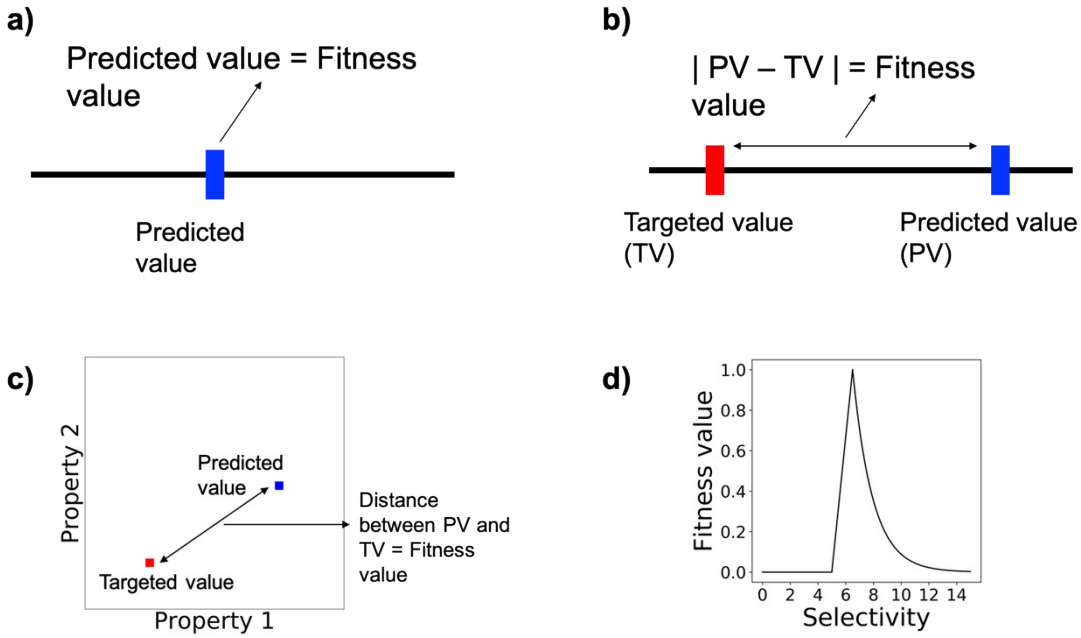

Fig. S6 Cost functions of genetic algorithms. a) Cost function to generate record-breaking MOFs. b) Cost function to generate MOFs which satisfy one kind of user-desired property. c) Cost function to generate MOFs which satisfy two different kinds of user-desired properties simultaneously. d) Cost function to generate MOFs which satisfy specific minimum selectivity value.

S4. Additional results for user-desired frameworks

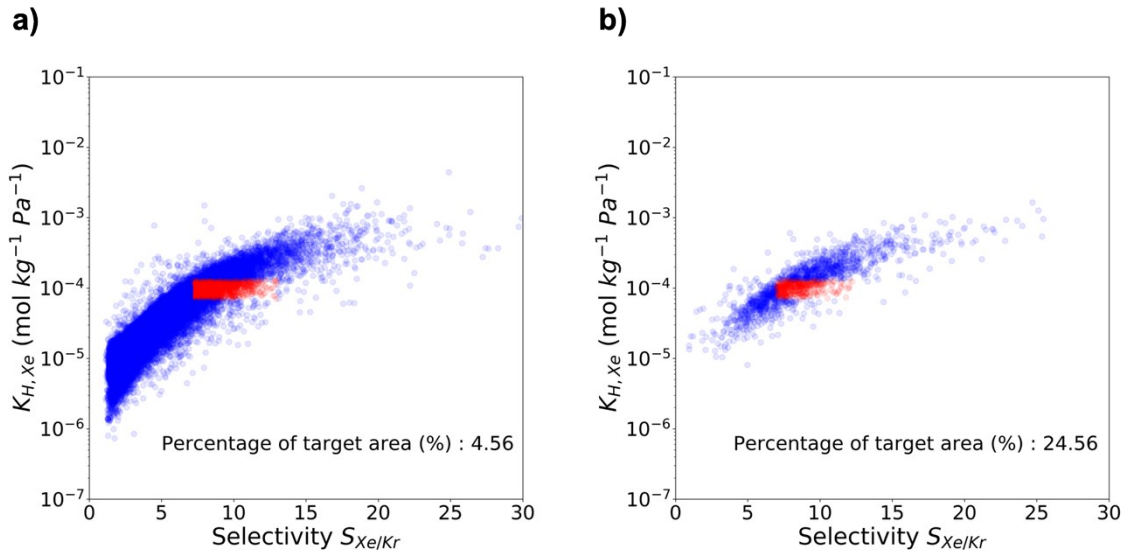

Fig. S7 Comparing the proportion of frameworks in the vicinity of target value of properties ($S_{Xe/Kr} = 10$, $K_{H,\,Xe} = 10^{-4}$). a) Selectivity versus $K_{H,\,Xe}$ plot of initial randomly generated frameworks. b) Selectivity versus $K_{H,\,Xe}$ plot of frameworks which were generated by genetic algorithm to target specific condition. Red area is $\pm$ 30% of target selectivity and $K_{H,Xe}$. There was obvious increase in proportion from 4.56 to 24.56 when we did inverse design.

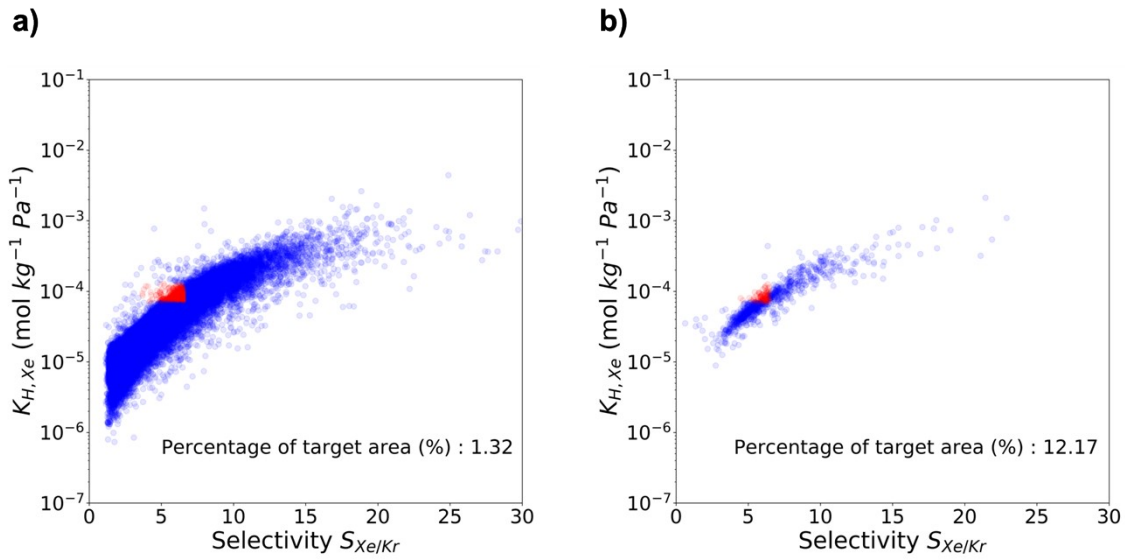

Fig. S8 Comparing the proportion of frameworks in the vicinity of target value of properties ($S_{Xe/Kr} = 5$, $K_{H,Xe} = 10^{-4}$). a) Selectivity versus $K_{H,\ Xe}$ plot of initial randomly generated frameworks. b) Selectivity versus $K_{H,\ Xe}$ plot of frameworks which were generated by genetic algorithm to target specific condition. Red area is $\pm$ 30% of target selectivity and $K_{H,Xe}$. There was obvious increase in proportion from 1.32 to 12.17 when we did inverse design.

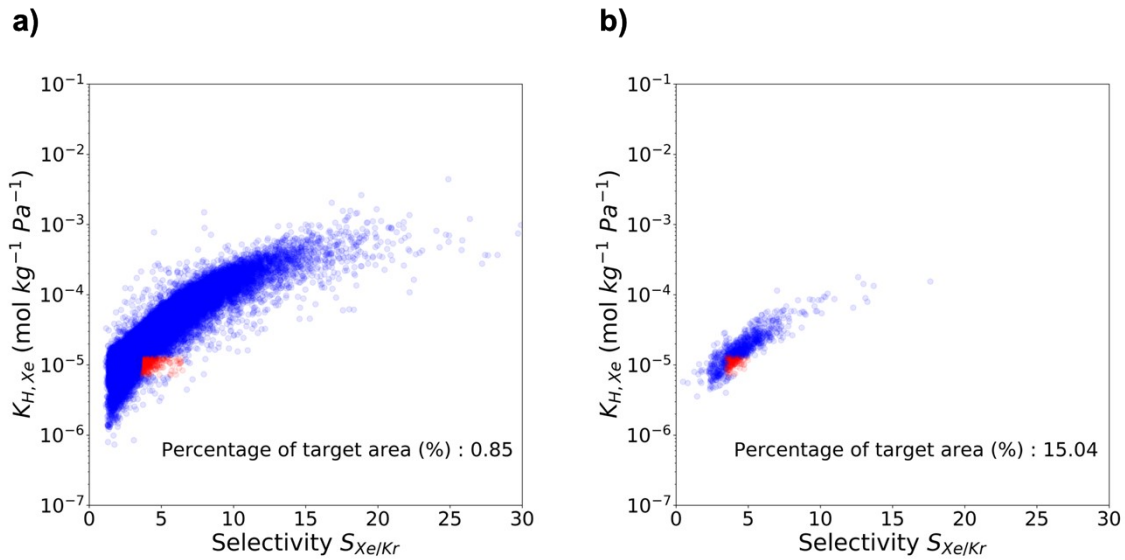

Fig. S9 Comparing the proportion of frameworks in the vicinity of target value of properties ($S_{Xe/Kr} = 5$, $K_{H,Xe} = 10^{-5}$). a) Selectivity versus $K_{H,\ Xe}$ plot of initial randomly generated frameworks. b) Selectivity versus $K_{H,\ Xe}$ plot of frameworks which were generated by genetic algorithm to target specific condition. Red area is $\pm$ 30% of target selectivity and $K_{H,Xe}$. There was obvious increase in proportion from 0.85 to 15.04 when we did inverse design.

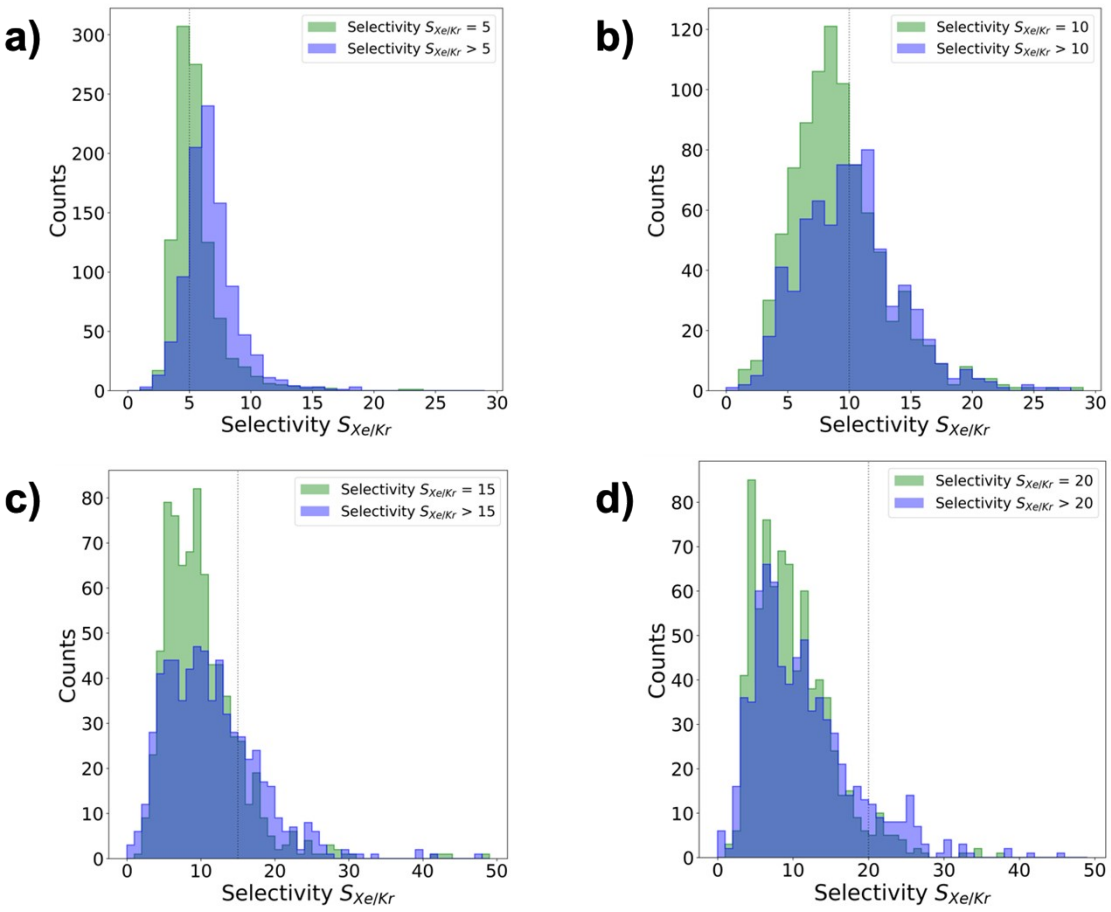

Fig. S10 Comparing distribution of selectivity of frameworks generated from the cost function to target specific selectivity and to target fine-tuned condition (minimum selectivity). a) Selectivity as 5. b) Selectivity as 10. c) Selectivity as 15. d) Selectivity as 20. Green histograms denote the results of the cost function to optimize frameworks to have specific targeted value of selectivity. Blue histograms denote the results of the cost function to optimize frameworks to have higher selectivity value than minimum threshold.

S5. Details on polymorph finder

Polymorph finder is polymorphic structure generation code to work with PORMAKE database.[6] As written in method section of manuscript, polymorph finder analyzes "connectivity" of frameworks to discriminate polymorphic candidates from all possible frameworks which can be made from same building blocks (node building blocks and edge building blocks). Connectivity means that how nodes are connected to each other. For convenience, we classified every topology in PORMAKE database into 186 groups by connectivity distribution of topologies (**Fig. S11**).

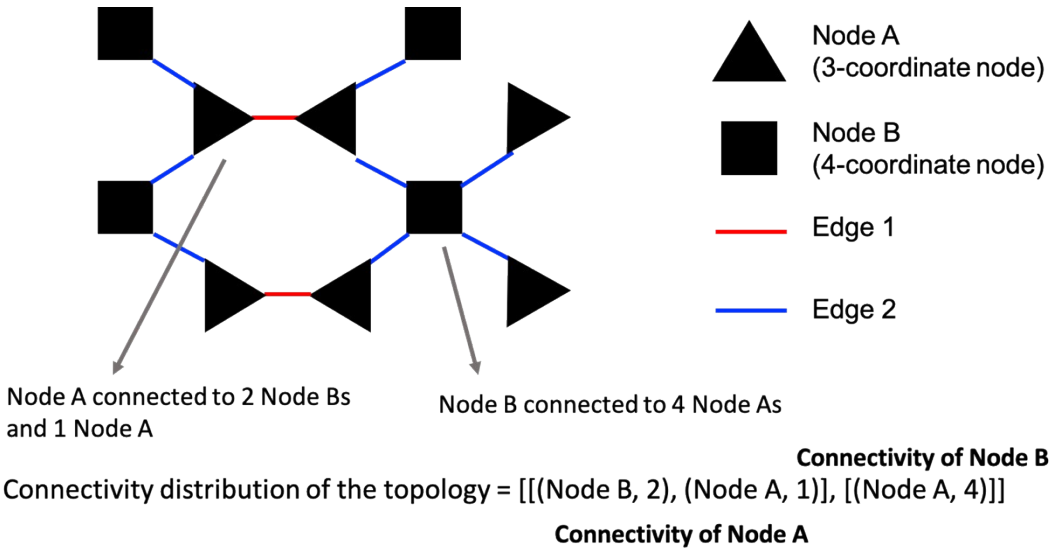

Fig. S11. Connectivity and Connectivity distribution of topology.

Polymorph finder gets a MOF as input. MOF should be converted into unique MOF representation, topology + NBBs + EBBs, to get overall data (topology and building blocks) of the framework properly. Once a unique MOF representation is given, the code identifies the possible topologies from PORMAKE database by analyzing connectivity distribution of the topology of input MOF. Then, for all possible topologies, every

combination of given building blocks (from input MOF) are generated. Finally, candidates, which have same ratio of building blocks combinations (NBB-EBB-NBB) with given MOF, are selected as polymorphic structures (**Fig. S12**).

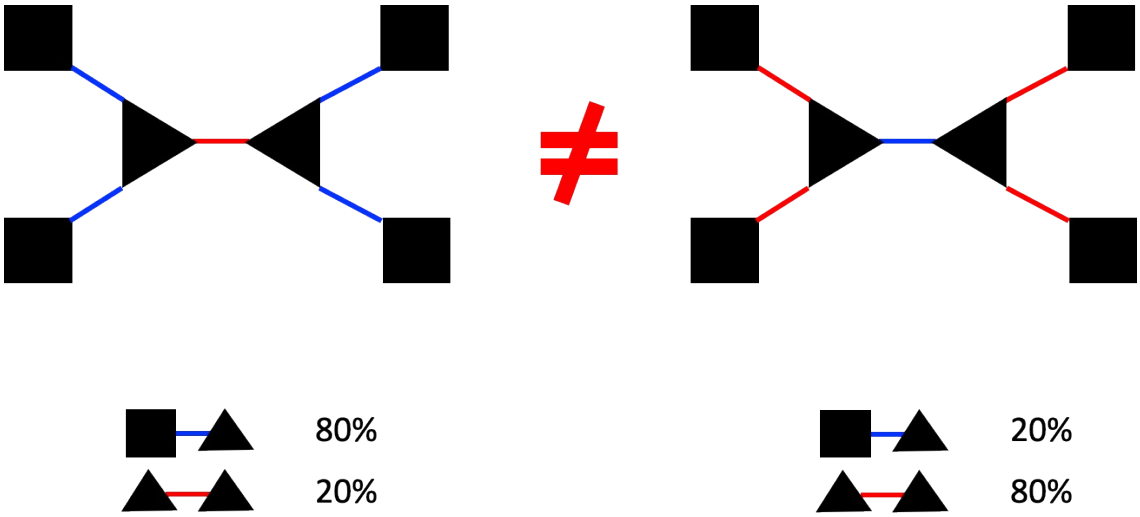

Fig. S12 Frameworks with different ratio of building block combinations are not regarded as polymorphic structures.

S6. Details on Monte Carlo simulation and additional candidates

As written in the manuscript, there are large discrepancy between experimental and theoretical results especially for optimal materials. Therefore, similar to the previous work, we did a literature survey to identify relationship between experimental and simulation data.[7-10] The exponential trend line can represent the relationship well and show that computational simulations can predict the trend of the optimal materials (**Fig. S13**).

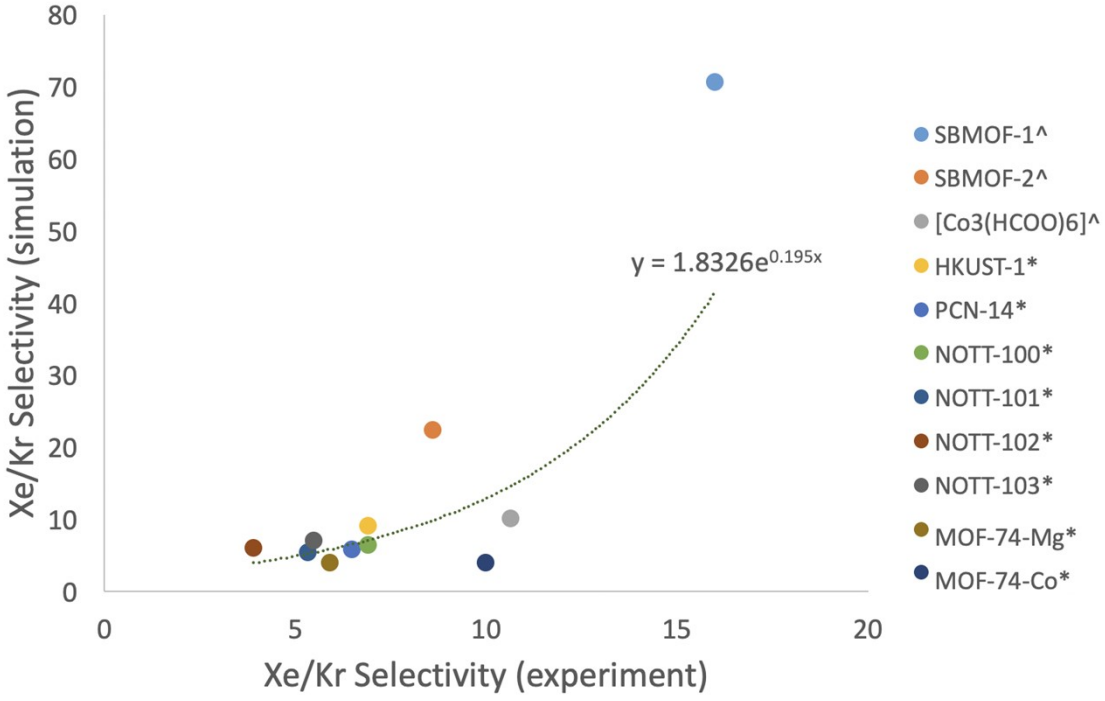

Fig. S13 Simulation results versus experimental values scatter plot for 11 structures from literature survey. The trend line is represented as exponential function.

(^ : at 298 K, * : at 292 K)

	ε/k_B **(K)**	σ (Å)		ε/k_B **(K)**	σ (Å)
Zn	62.4	2.46	**Se**	146.4177558	3.7462345

C	52.84	3.43	**Rb**	20.126152	3.6651626
H	22.14	2.57	**Sr**	118.241143	3.2437669
O	30.19	3.12	**Y**	36.2270736	2.9800605
Mg	55.85	2.69	**Ru**	28.1766128	2.6397367
N	34.72	3.26	**Pd**	24.1513824	2.5827191
S	137.89	3.6	**Sn**	285.2882046	3.9128328
Cl	114.24	3.516	**Sb**	225.9160562	3.937778
F	25.163	2.996	**Ba**	183.1479832	3.2990027
Br	126.32	3.731	**Pr**	5.031538	3.2125854
Cu	2.515	3.113	**Nd**	5.031538	3.1849675
V	8.05	2.801	**Sm**	4.0252304	3.135968
Zr	34.72	2.78	**Tb**	3.5220766	3.0744959
Si	202.3	3.825	**Dy**	3.5220766	3.0540052
I	170.6	4	**Ho**	3.5220766	3.0370781
Ge	190.71	3.812	**Er**	3.5220766	3.01985
P	153.47	3.693	**Tm**	3.0189228	3.0058966
Co	7.044	2.5576	**Yb**	114.7190664	2.9889695
Mn	6.541	2.638	**W**	33.7113046	2.7341721
Eu	4.025	3.1119	**Re**	33.2081508	2.6317186
Ag	18.1135	2.804	**Au**	19.6229982	2.9337337
Gd	4.528	3	**Hg**	193.714213	2.4098845
Cs	22.64	4.024	**Pb**	333.5909694	3.8281973
In	301.38	3.976	**Bi**	260.6336684	3.893233

Mo	28.176	2.719	**Th**	13.0819988	3.0254964
Ce	6.541	3.168	**U**	11.0693836	3.0246055
Cd	114.72	2.5372	**Lu**	20.6293	3.2428
Nb	29.686	2.819	**Pt**	40.2523	2.453539
Na	15.094	2.657	**Te**	200.25	3.9823
La	8.553	3.1377	**Be**	42.768	2.4455
Li	12.578845	2.1835959	**Pu**	8.05046	3.0504
B	90.567684	3.6375447	**Ti**	8.554	2.829
Al	254.092669	4.0081591	**Ir**	36.719	2.492
K	17.610383	3.3961108	**Os**	18.617	2.778
Ca	119.7506044	3.0281691	**Ta**	40.757	2.823
Sc	9.5599222	2.9355155	**Rh**	26.668	2.6084
Cr	7.547307	2.6931907	**Am**	7.044	3.0109
Fe	6.5409994	2.5943008	**Np**	9.56	3.049
Ni	7.547307	2.5248106	**Tl**	342.16	3.871
Ga	208.808827	3.9048147	**Hf**	36.23	2.797
As	155.4745242	3.768507			

Table S4. Universial Force Field (UFF) parameters for host frameworks, MOFs.

	ε/k_B **(K)**	σ **(Å)**
Xenon	229.8	3.97
Krypton	165.2	3.66

Table S5. Force field parameters for guest atoms, Xenon and Krypton

	$K_{H,Xe}$ (mol kg^{-1} Pa^{-1})	$S_{Xe/Kr}$	LCD (Å)	PLD (Å)
htp+N5+N92+E0	5.8×10^{-2}	**78**	**5.1**	**4.6**
htp+N5+N270+E0	1.6×10^{-1}	**100**	**5.0**	**4.5**
pha+N271+N328+E127+E0	9.4×10^{-3}	77	25	17
pha+N271+N328+E159+E0	5.9×10^{-3}	71	29	20.
cdp+N164+N693+E0+E0	5.4×10^{-2}	73	4.8	3.5
xik+N388+N688+E0+E0	7.7×10^{-2}	99	6.9	3.8
whq+N270+E0	5.2×10^{-2}	110	9.5	7.6
fte+N3+N238+E0+E0+E223	3.6×10^{-1}	290	8.0	3.5
fte+N3+N238+E0+E0+E23	2.9×10^{-1}	250	5.8	3.5
fte+N3+N238+E0+E0+E156	2.7×10^{-1}	230	4.6	3.2
fjh+N130+N2+N223+E0+E0	2.5×10^{-2}	76	5.5	3.9
pha+N271+N328+E88+E0	9.5×10^{-3}	77	24	16
urt+N544+N522+N495+E79+E0	7.7×10^{-2}	100	8.3	3.7
bbd+N434+N223+E184+E25+E0	1.1×10^{-2}	77	15	10.
bbd+N434+N223+E0+E25+E0	1.0×10^{-2}	78	15	10.
bbd+N434+N223+E0+E205+E0	1.7×10^{-2}	92	14	8.6

Table S6. Summary on 16 candidates after flexible molecular simulation and RASPA simulation. Final 2 candidates after polymorphic simulations are highlighted as bold type. Xenon Henry coefficient and Xe/Kr selectivity were calculated via in-house GPU codes and geometric descriptors (LCD and PLD) were calculated via Zeo++ software.

References

1. J. Kim, R. L. Martin, O. Rübel, M. Haranczyk and B. Smit, *Journal of Chemical Theory and Computation*, 2012, **8**, 1684-1693.
2. D. Dubbeldam, S. Calero, D. E. Ellis and R. Q. Snurr, *Molecular Simulation*, 2016, **42**, 81-101.
3. K. M. L. Taylor, A. Jin and W. Lin, *Angew. Chem., Int. Ed.*, 2008, **47**, 7722.
4. S.-T. Zheng, J. T. Bu, Y. Li, T. Wu, F. Zuo, P. Feng and X. Bu, *Journal of the American Chemical Society*, 2010, **132**, 17062-17064.
5. C. Volkringer, T. Loiseau, G. Férey, C. M. Morais, F. Taulelle, V. Montouillout and D. Massiot, *Microporous and Mesoporous Materials*, 2007, **105**, 111-117.
6. S. Lee, B. Kim, H. Cho, H. Lee, S. Y. Lee, E. S. Cho and J. Kim, *ACS Applied Materials & Interfaces*, 2021, **13**, 23647-23654.
7. D. Banerjee, C. M. Simon, A. M. Plonka, R. K. Motkuri, J. Liu, X. Chen, B. Smit, J. B. Parise, M. Haranczyk and P. K. Thallapally, *Nature Communications*, 2016, **7**, ncomms11831.
8. X. Chen, A. M. Plonka, D. Banerjee, R. Krishna, H. T. Schaef, S. Ghose, P. K. Thallapally and J. B. Parise, *Journal of the American Chemical Society*, 2015, **137**, 7007-7010.
9. H. Wang, K. Yao, Z. Zhang, J. Jagiello, Q. Gong, Y. Han and J. Li, *Chemical Science*, 2014, **5**, 620-624.
10. J. J. Perry, S. L. Teich-McGoldrick, S. T. Meek, J. A. Greathouse, M. Haranczyk and M. D. Allendorf, *The Journal of Physical Chemistry C*, 2014, **118**, 11685-11698.