A machine-learning view on heterogeneous catalyst design and discovery

Ichigaku Takigawa
ichigaku.takigawa@riken.jp

Telluride Workshop on Computational Materials Chemistry
1 July 2021 @ Telluride, Colorado

RIKEN AIP Advanced Intelligence Project

ICReDD Institute for Chemical Reaction Design and Discovery HOKKAIDO UNIVERSITY

Hi, I am a ML researcher working for

ML for Stem Cell Biology

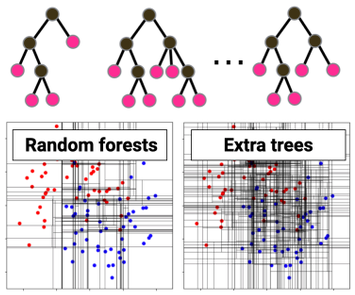
RIKEN Center for AI Project

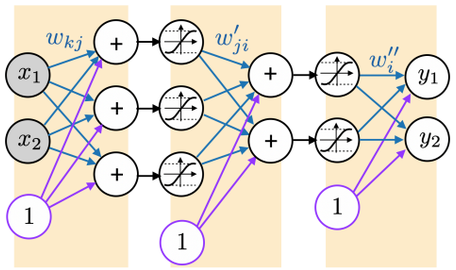
ML for Chemistry

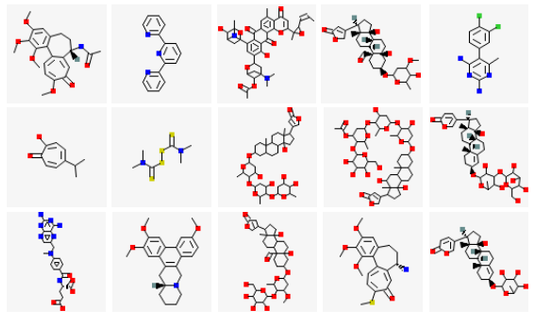
Inst. Chemical Reaction Design & Discovery
Hokkaido Univ

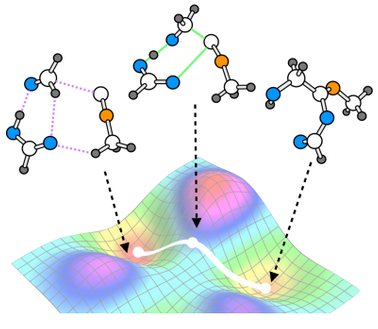
Two Interrelated Research Interests:

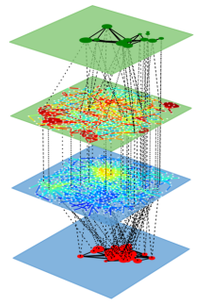
ML with **discrete (combinatorial) structures** -------- ML for **natural sciences**

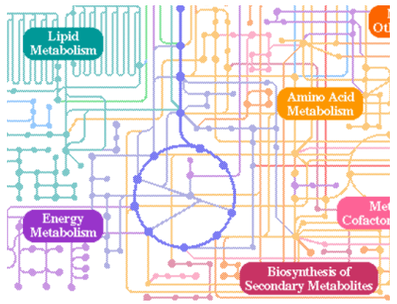




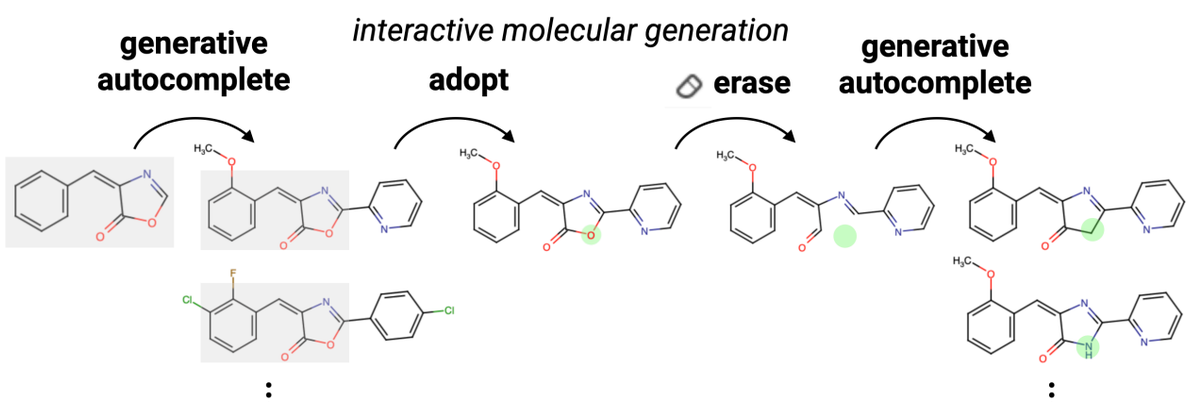




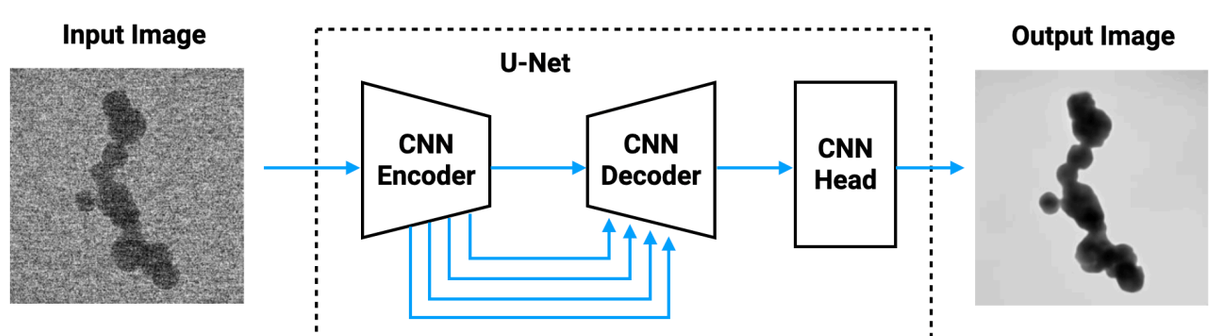




Edit-aware graph autocompletion (Hu+)



Low-electron dose TEM image improvement (Katsuno+)



Today's talk

Our struggles for better ML practices with underspecified, sparse, biased observational data (i.e. a collection of experimental facts from literature)

ChemCatChem
The European Society Journal for Catalysis
Chemistry Europe
European Chemical Societies Publishing

Analysis of Updated Literature Data up to 2019 on the Oxidative Coupling of Methane Using an Extrapolative Machine-Learning Method to Identify Novel Catalysts

Dr. Shinya Mine, Motoshi Takao, Taichi Yamaguchi, Dr. Takashi Toyao ✉, Dr. Zen Maeno, Dr. S. M. A. Hakim Siddiki, Dr. Satoru Takakusagi, Prof. Ken-ichi Shimizu ✉, Dr. Ichigaku Takigawa ✉

First published: 31 May 2021 | https://doi.org/10.1002/cctc.202100495

ChemCatChem
The European Society Journal for Catalysis
Chemistry Europe
European Chemical Societies Publishing

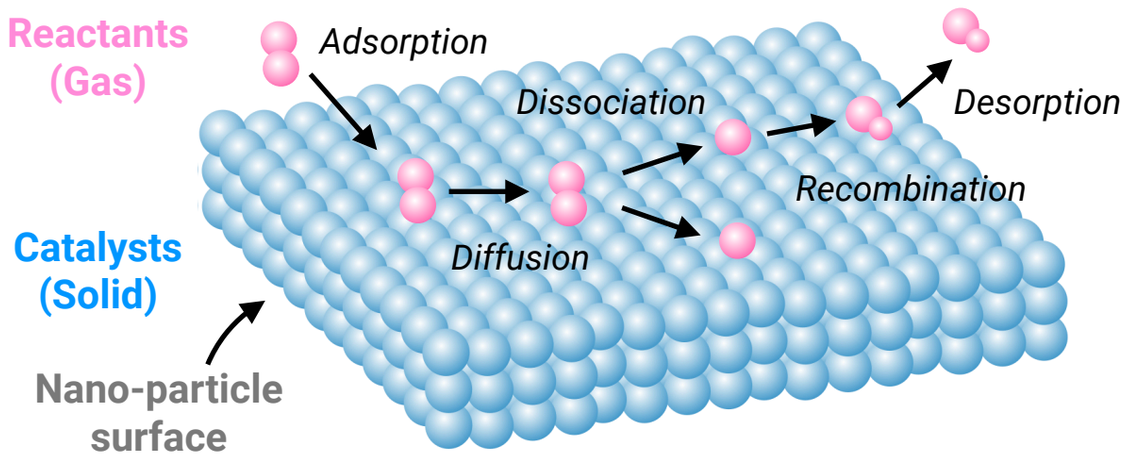
Statistical Analysis and Discovery of Heterogeneous Catalysts Based on Machine Learning from Diverse Published Data

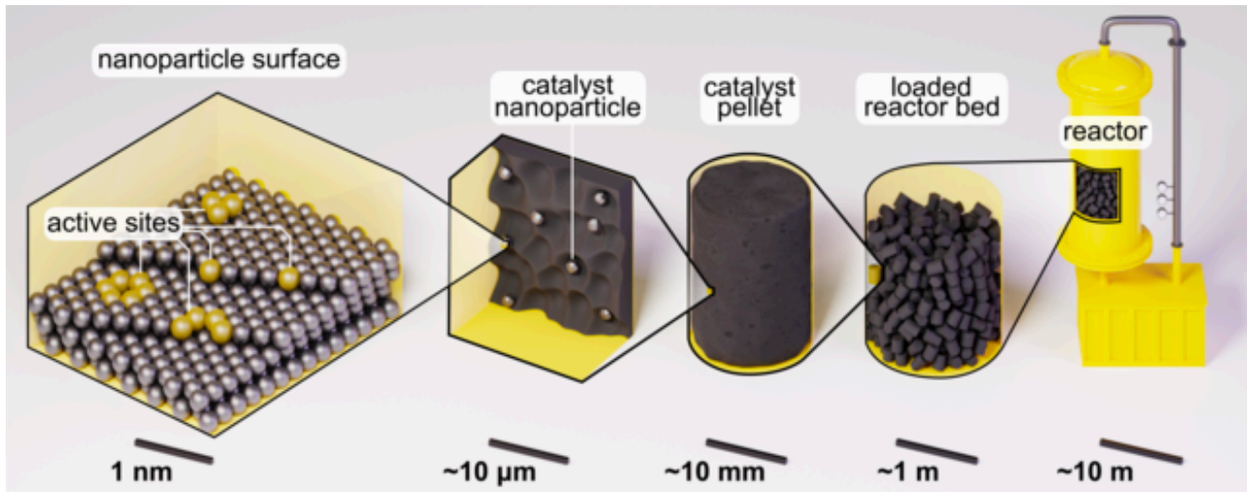
Keisuke Suzuki, Dr. Takashi Toyao, Dr. Zen Maeno, Dr. Satoru Takakusagi, Prof. Ken-ichi Shimizu ✉, Dr. Ichigaku Takigawa ✉

First published: 09 July 2019 | https://doi.org/10.1002/cctc.201900971 | Citations: 10

Gas-phase reactions on sold-phase catalyst surface (Heterogeneous catalysis)

Industrial Synthesis (e.g. Haber-Bosch), Automobile Exhaust Gas Purification, Methane Conversion, etc.





https://en.wikipedia.org/wiki/Heterogeneous_catalysis

Today's talk

Our struggles for better ML practices with underspecified, sparse, biased observational data (i.e. a collection of experimental facts from literature)

ChemCatChem
The European Society Journal for Catalysis
Chemistry Europe
European Chemical Societies Publishing

Analysis of Updated Literature Data up to 2019 on the Oxidative Coupling of Methane Using an Extrapolative Machine-Learning Method to Identify Novel Catalysts

Dr. Shinya Mine, Motoshi Takao, Taichi Yamaguchi, Dr. Takashi Toyao ✉, Dr. Zen Maeno, Dr. S. M. A. Hakim Siddiki, Dr. Satoru Takakusagi, Prof. Ken-ichi Shimizu ✉, Dr. Ichigaku Takigawa ✉

First published: 31 May 2021 | https://doi.org/10.1002/cctc.202100495

ChemCatChem
The European Society Journal for Catalysis
Chemistry Europe
European Chemical Societies Publishing

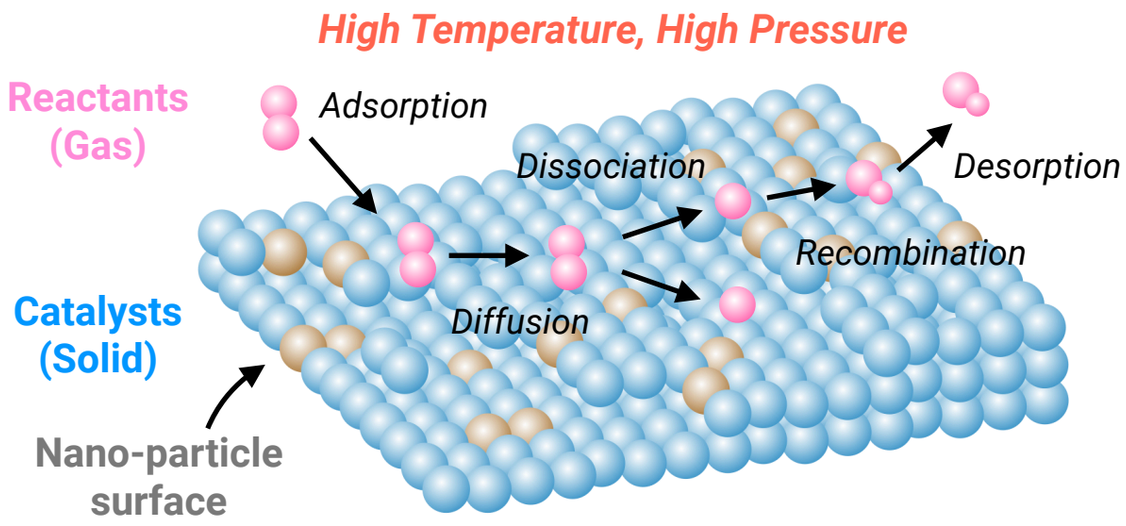
Statistical Analysis and Discovery of Heterogeneous Catalysts Based on Machine Learning from Diverse Published Data

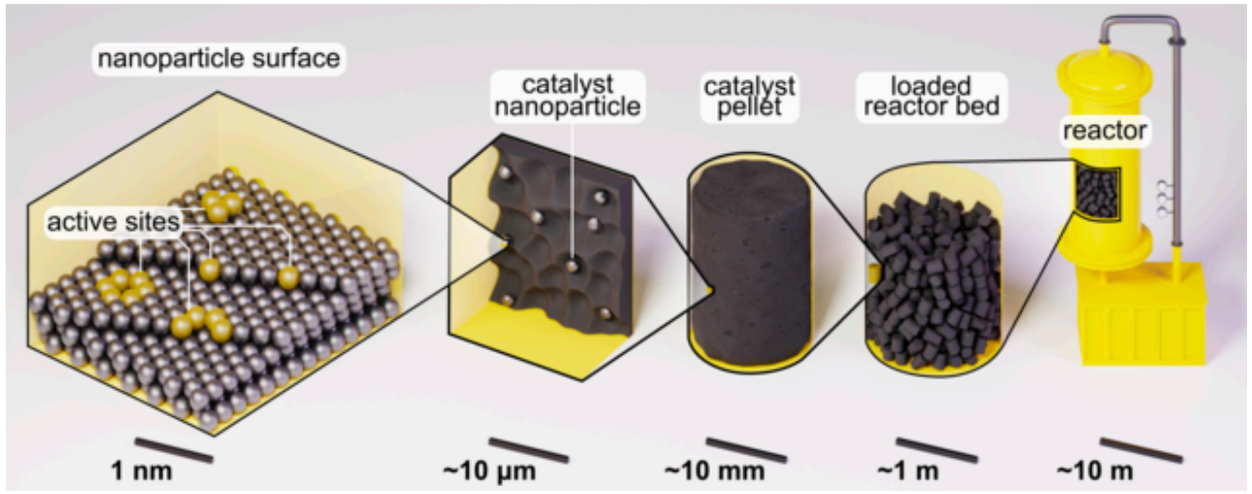
Keisuke Suzuki, Dr. Takashi Toyao, Dr. Zen Maeno, Dr. Satoru Takakusagi, Prof. Ken-ichi Shimizu ✉, Dr. Ichigaku Takigawa ✉

First published: 09 July 2019 | https://doi.org/10.1002/cctc.201900971 | Citations: 10

Gas-phase reactions on sold-phase catalyst surface (Heterogeneous catalysis)

Industrial Synthesis (e.g. Haber-Bosch), Automobile Exhaust Gas Purification, Methane Conversion, etc.





https://en.wikipedia.org/wiki/Heterogeneous_catalysis

Today's talk

Our struggles for better ML practices with underspecified, sparse, biased observational data (i.e. a collection of experimental facts from literature)

ChemCatChem
The European Society Journal for Catalysis
Chemistry Europe
European Chemical Societies Publishing

Analysis of Updated Literature Data up to 2019 on the Oxidative Coupling of Methane Using an Extrapolative Machine-Learning Method to Identify Novel Catalysts

Dr. Shinya Mine, Motoshi Takao, Taichi Yamaguchi, Dr. Takashi Toyao ✉, Dr. Zen Maeno, Dr. S. M. A. Hakim Siddiki, Dr. Satoru Takakusagi, Prof. Ken-ichi Shimizu ✉, Dr. Ichigaku Takigawa ✉

First published: 31 May 2021 | https://doi.org/10.1002/cctc.202100495

ChemCatChem
The European Society Journal for Catalysis
Chemistry Europe
European Chemical Societies Publishing

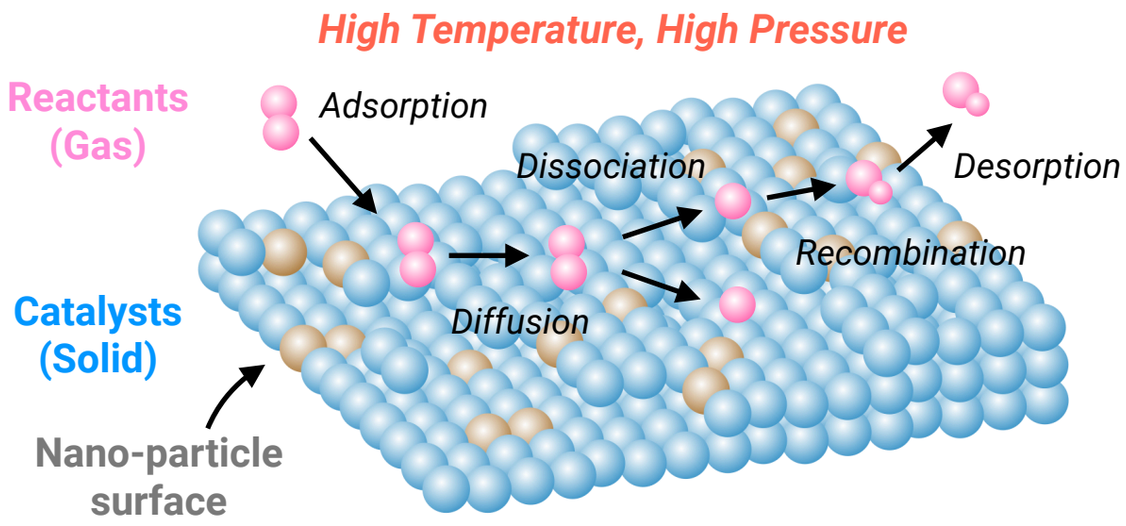
Statistical Analysis and Discovery of Heterogeneous Catalysts Based on Machine Learning from Diverse Published Data

Keisuke Suzuki, Dr. Takashi Toyao, Dr. Zen Maeno, Dr. Satoru Takakusagi, Prof. Ken-ichi Shimizu ✉, Dr. Ichigaku Takigawa ✉

First published: 09 July 2019 | https://doi.org/10.1002/cctc.201900971 | Citations: 10

Gas-phase reactions on sold-phase catalyst surface (Heterogeneous catalysis)

Industrial Synthesis (e.g. Haber-Bosch), Automobile Exhaust Gas Purification, Methane Conversion, etc.



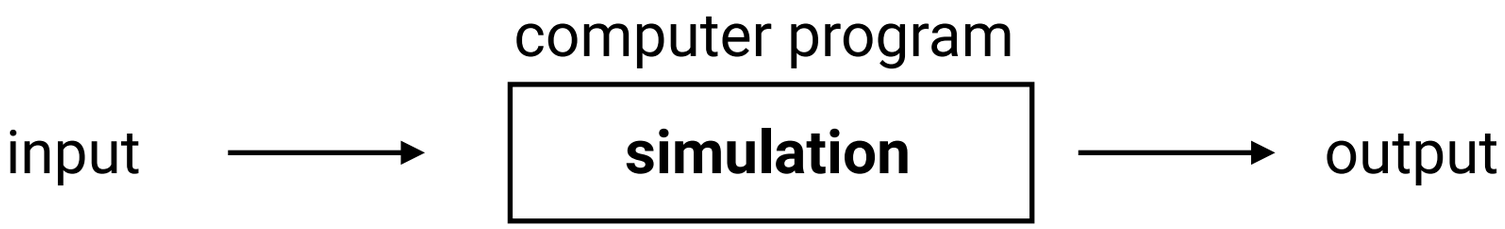
Devilishly complex too-many-factor process!!

God made the bulk;
the **surface** was invented by the **devil**

—— Wolfgang Pauli

ML: A new way for (lazy) programming

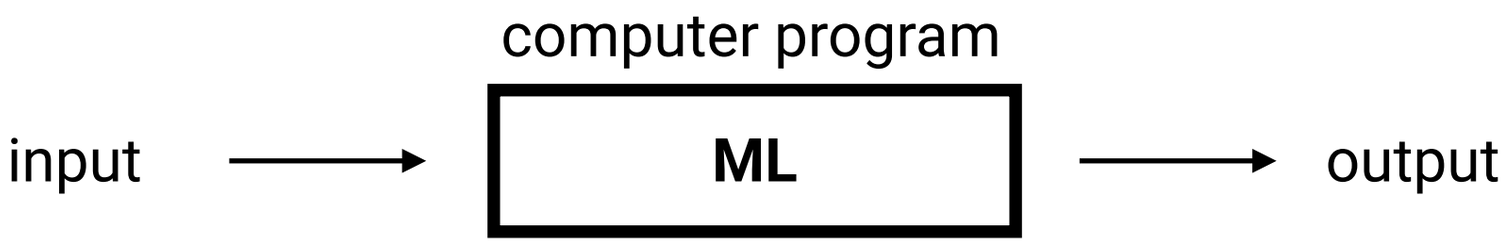
deductive (rationalism)



full specification in every detail required

---

inductive (empiricism)

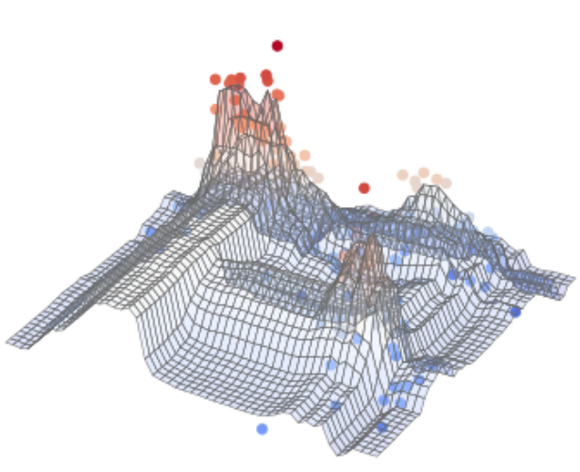


airhead with a god-like learning capability

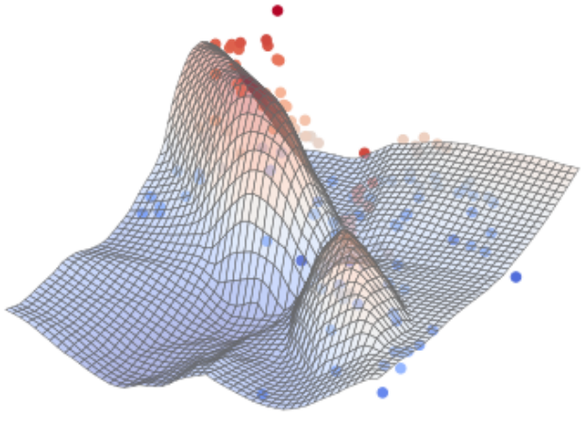
give up explicit model

instead, grab a tunable model, and show it many input-output instances

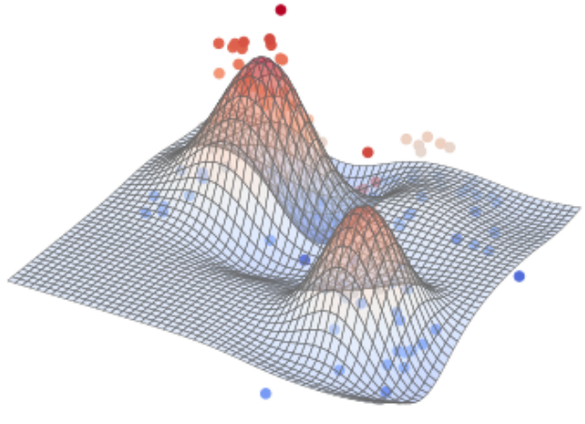
All about fitting a **very-flexible** function to **finite** points in **high-dimensional** space.



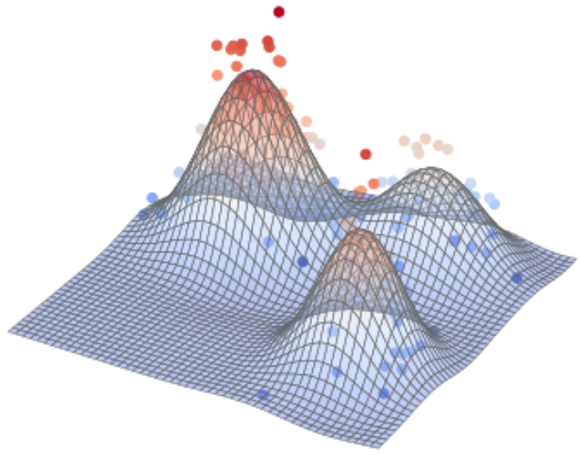
Random Forest



Neural Networks



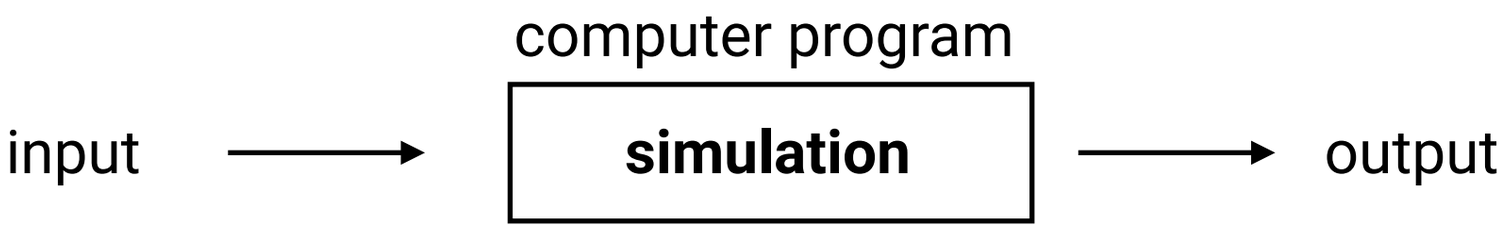
SVR



Kernel Ridge

ML: A new way for (lazy) programming

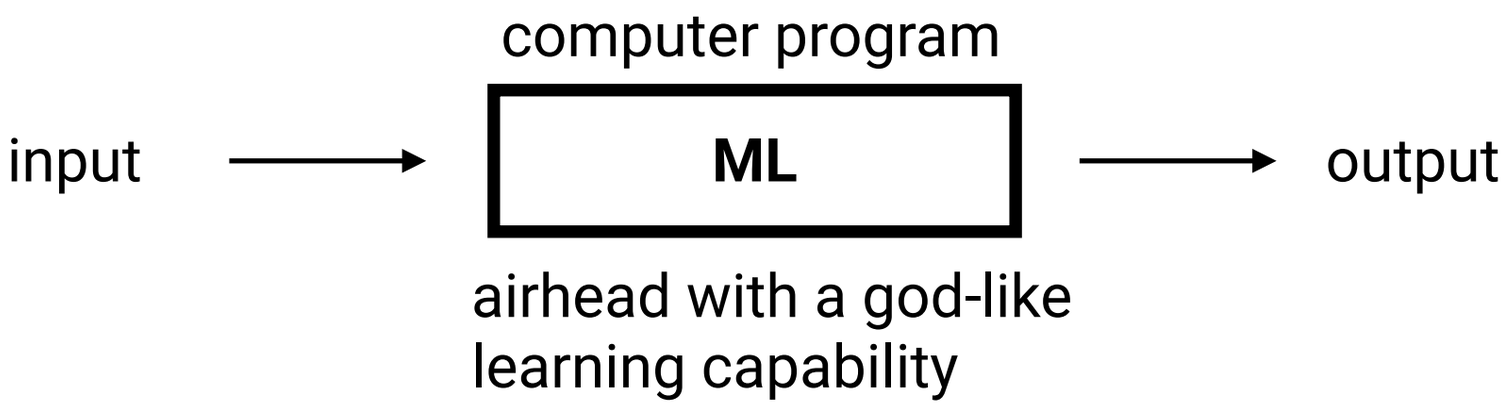
deductive (rationalism)



full specification in every detail required

---

inductive (empiricism)



give up explicit model

instead, grab a tunable model, and show it many input-output instances

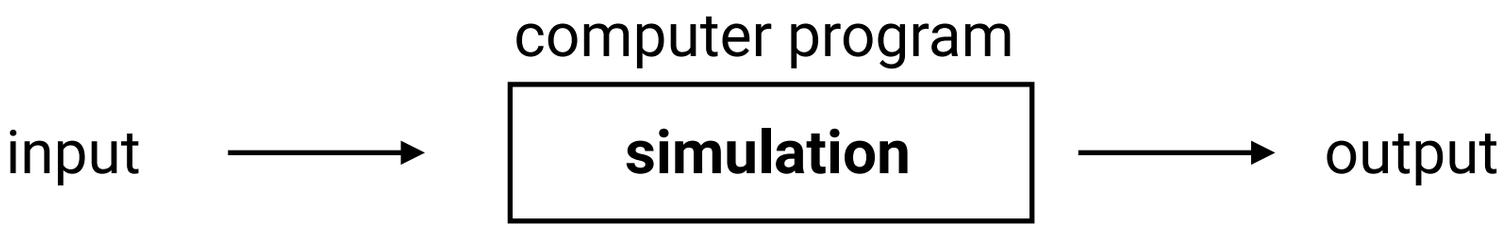
All about fitting a **very-flexible** function to **finite** points in **high-dimensional** space.

ResNet50: 26 million params
ResNet101: 45 million params
EfficientNet-B7: 66 million params
VGG19: 144 million params

12-layer, 12-heads BERT: 110 million params
24-layer, 16-heads BERT: 336 million params
GPT-2 XL: 1558 million params
GPT-3: 175 billion params

ML: A new way for (lazy) programming

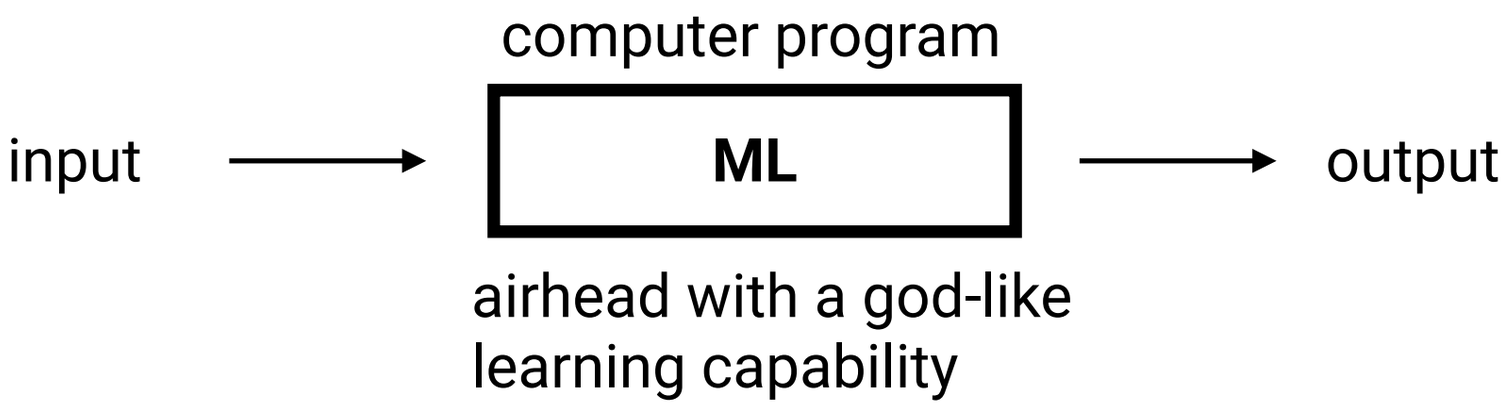
deductive (rationalism)



full specification in every detail required

---

inductive (empiricism)



give up explicit model

instead, grab a tunable model, and show it many input-output instances

All about fitting a **very-flexible** function to **finite** points in **high-dimensional** space.

ResNet50: 26 million params
ResNet101: 45 million params
EfficientNet-B7: 66 million params
VGG19: 144 million params

12-layer, 12-heads BERT: 110 million params
24-layer, 16-heads BERT: 336 million params
GPT-2 XL: 1558 million params
GPT-3: 175 billion params

Modern ML: Can we imagine what would happen if we try to fit a function having **175 billion** parameters to **100 million** data points in **10 thousand** dimension??

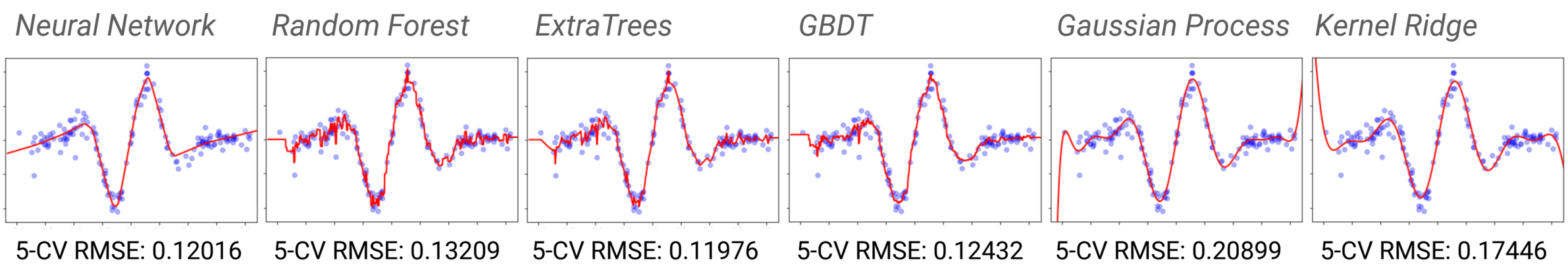
Rashomon Effect: multiplicity of good models

ML models are **too flexible to overrepresent given finite instances**, and many different shapes of functions exist for representing the same **finite** data. (even if it's huge)

The Rashomon Effect

In many practical cases, we have many equally-accurate but different ML models. (the choice of ML methods or the design of NN architectures doesn't really matter)

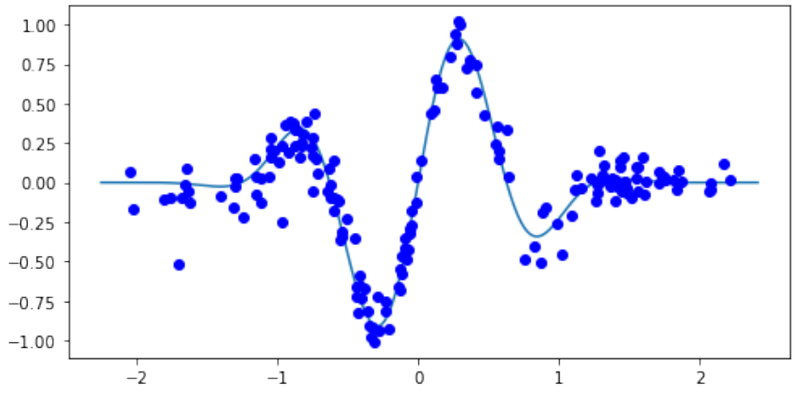
We often see this in ML competitions. Top ranking solutions are very competitive in performance (equally accurate in practice) but can be very different approaches.



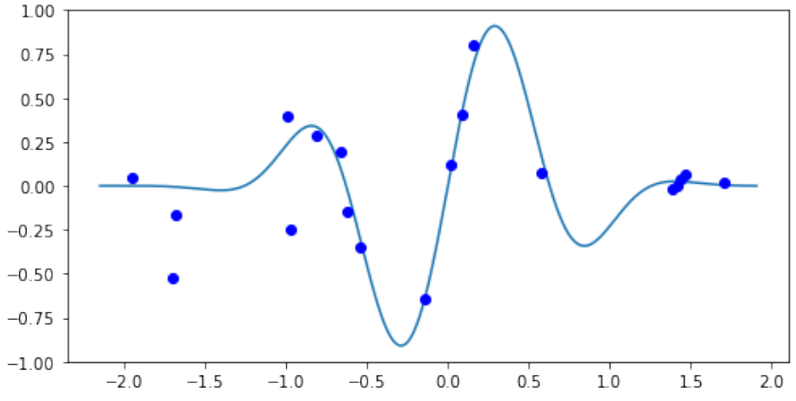
Note: The Rashomon Effect in ML is attributed to Leo Breiman's very influential "The Two Cultures" paper published in 2001, but obviously "Rashomon" itself originates from a classic Japanese movie in 1950 by Kurosawa, where four witnesses to a murder describe entirely different contradictory perspectives, but all of them sound true.

We see differences in underspecified cases

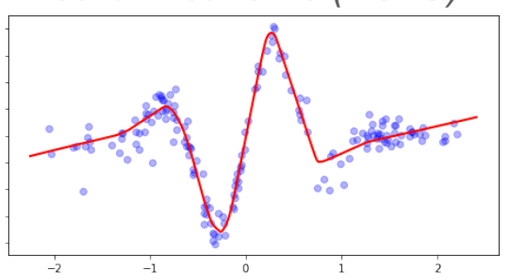
right data



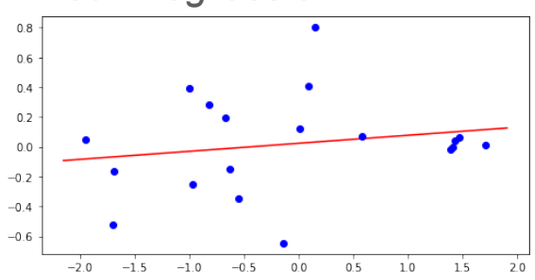
scarce/underspecified + outliers



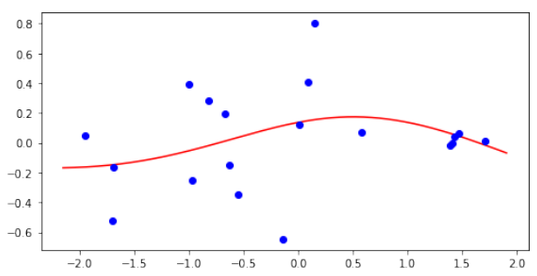
Neural Networks (ReLU)



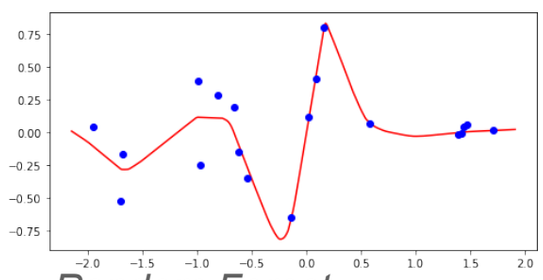
Linear Regression



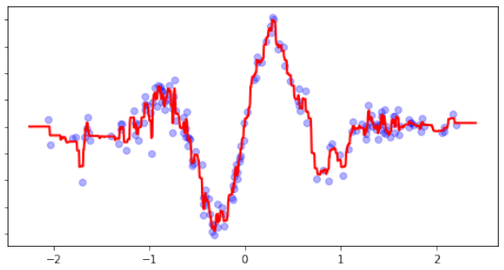
Neural Networks (Tanh)



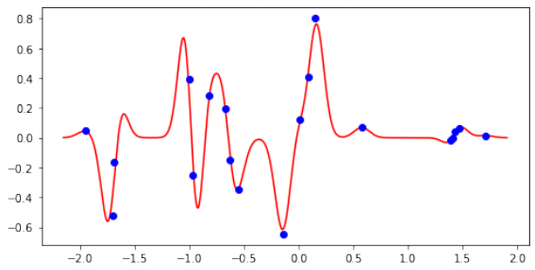
Neural Networks (ReLU)



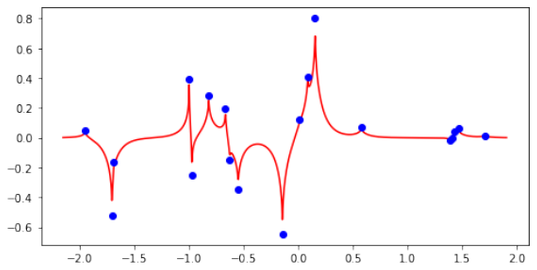
Random Forest



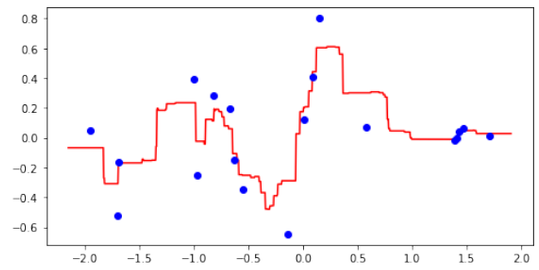
Kernel Ridge (RBF)



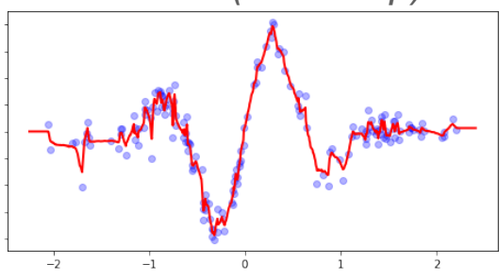
Kernel Ridge (Laplacian)



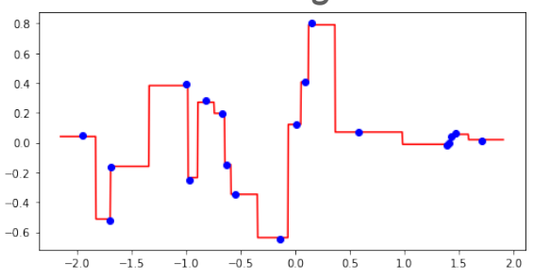
Random Forest



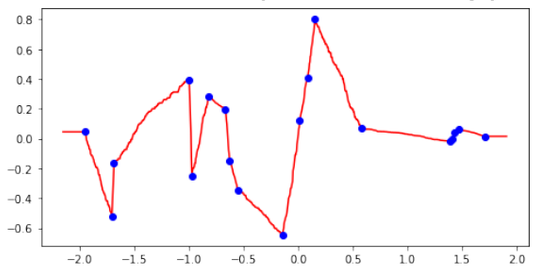
Extra Trees (bootstrap)



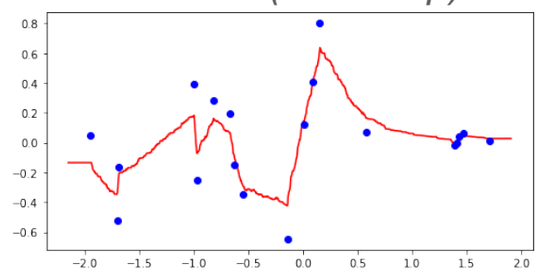
Gradient Boosting



Extra Trees (no bootstrap)

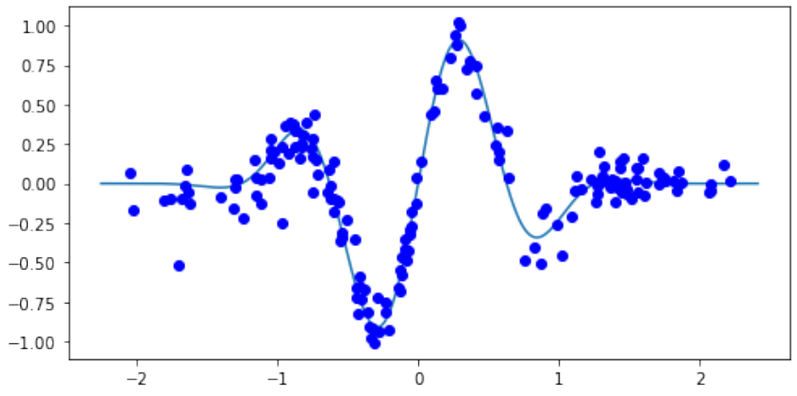


Extra Trees (bootstrap)

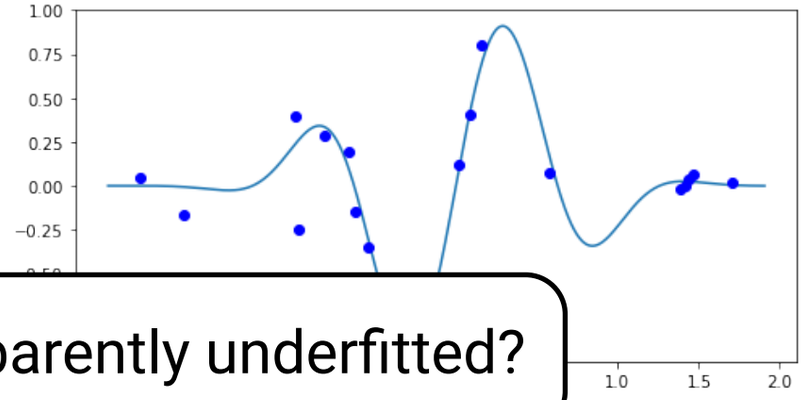


We see differences in underspecified cases

right data

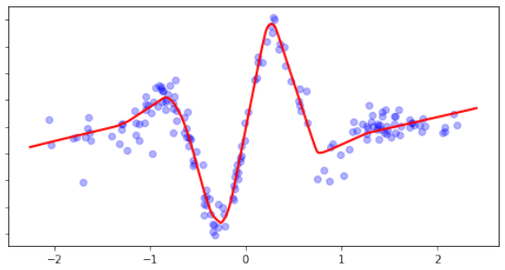


scarce/underspecified + outliers

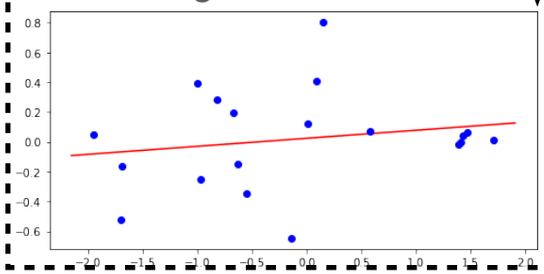


some apparently underfitted?

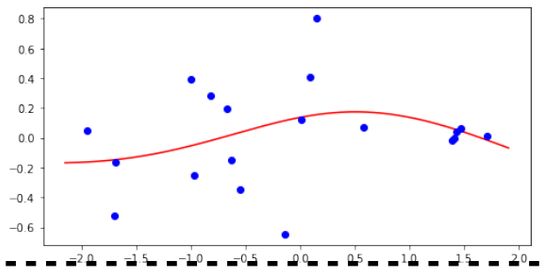
Neural Networks (ReLU)



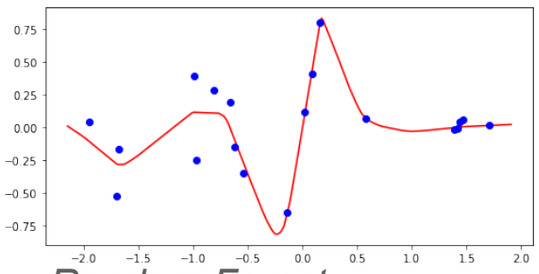
Linear Regression



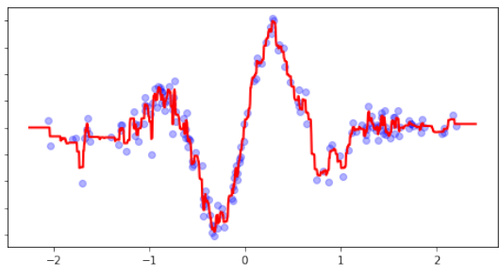
Neural Networks (Tanh)



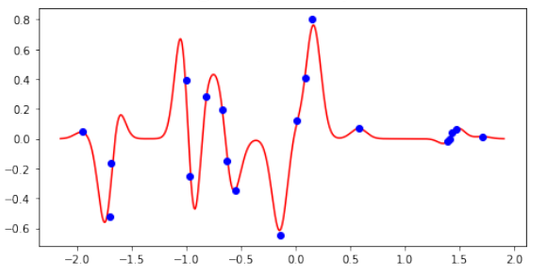
Neural Networks (ReLU)



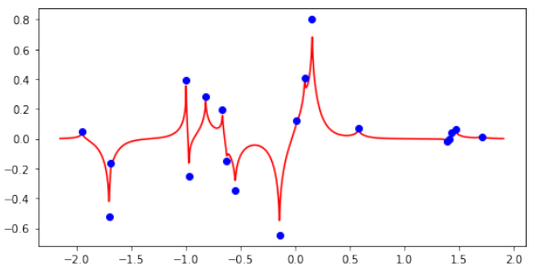
Random Forest



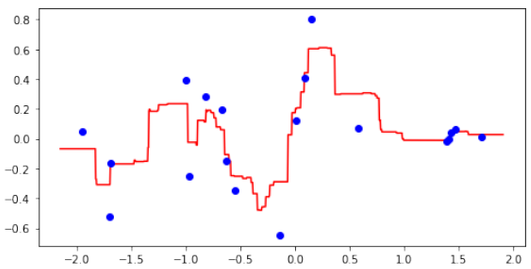
Kernel Ridge (RBF)



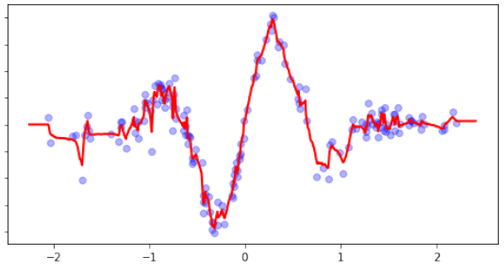
Kernel Ridge (Laplacian)



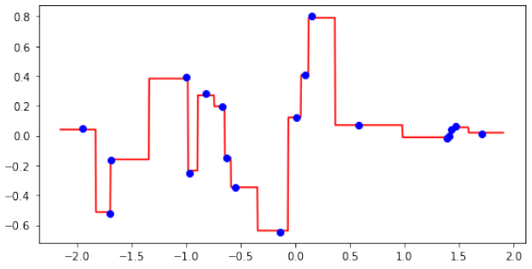
Random Forest



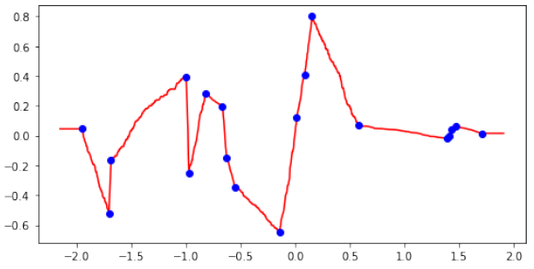
Extra Trees (bootstrap)



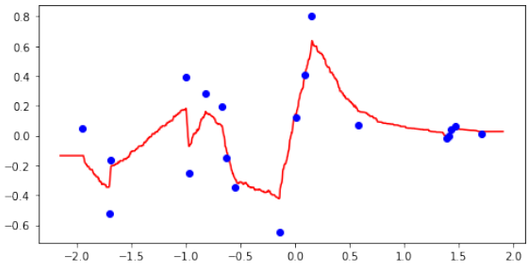
Gradient Boosting



Extra Trees (no bootstrap)

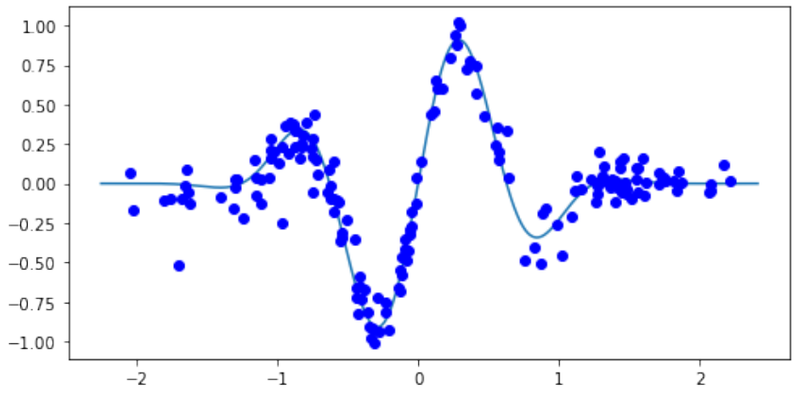


Extra Trees (bootstrap)

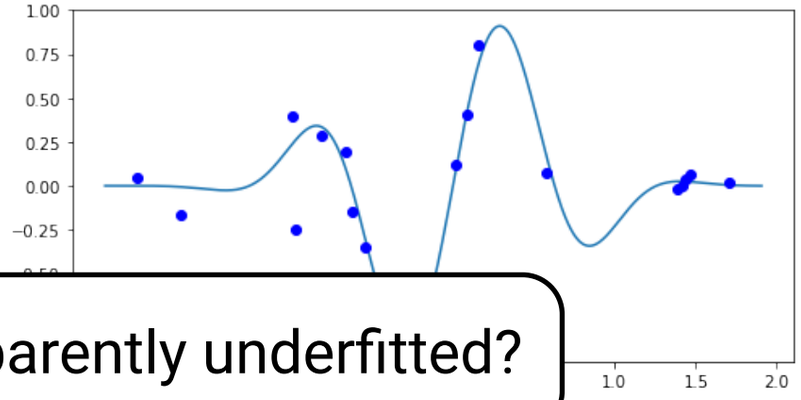


We see differences in underspecified cases

right data

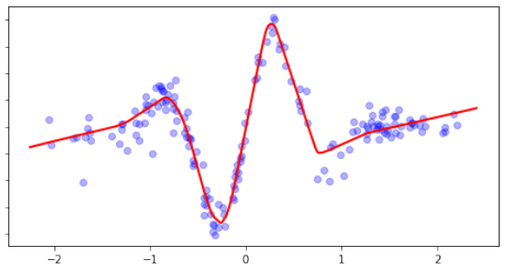


scarce/underspecified + outliers

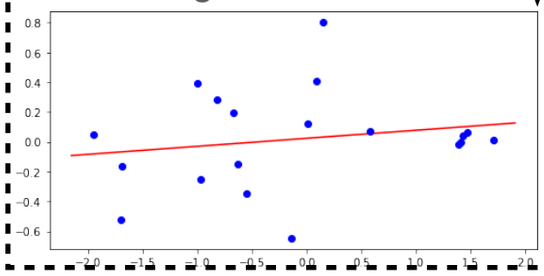


some apparently underfitted?

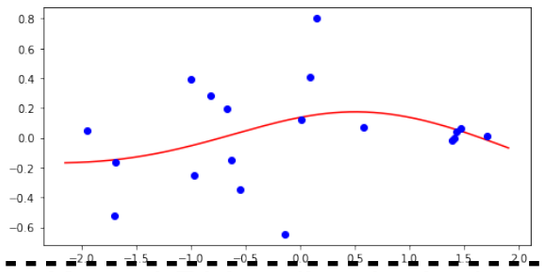
Neural Networks (ReLU)



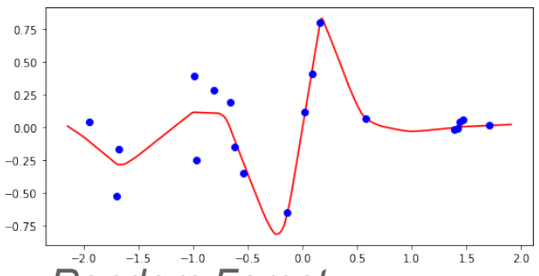
Linear Regression



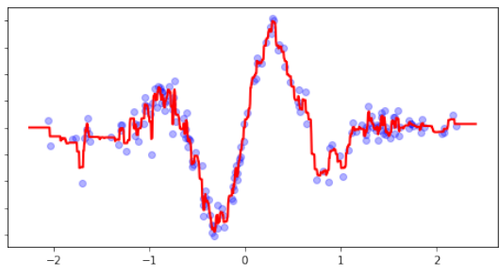
Neural Networks (Tanh)



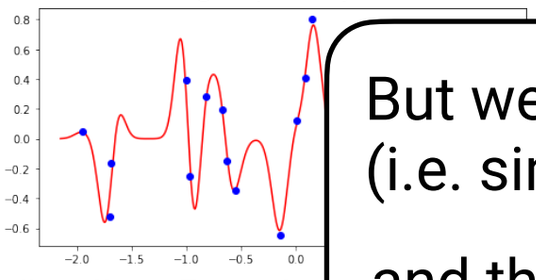
Neural Networks (ReLU)



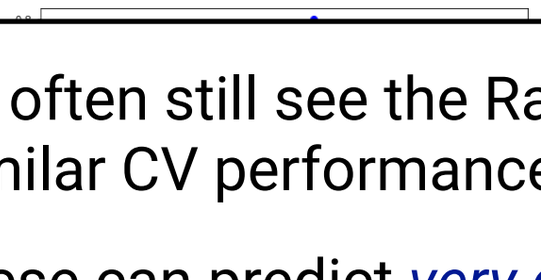
Random Forest



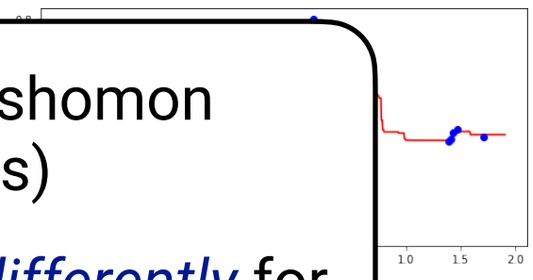
Kernel Ridge (RBF)



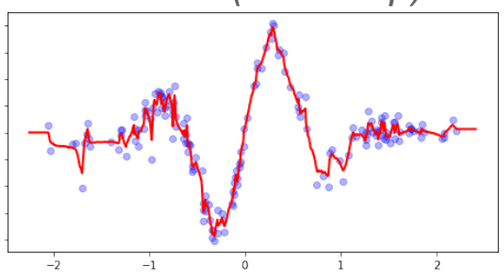
Kernel Ridge (Laplacian)



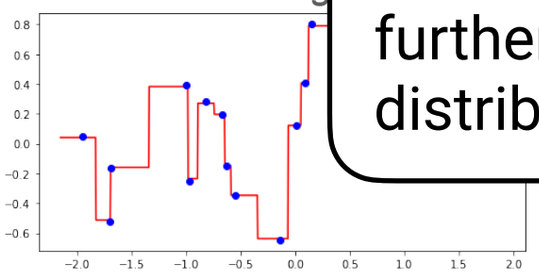
Random Forest

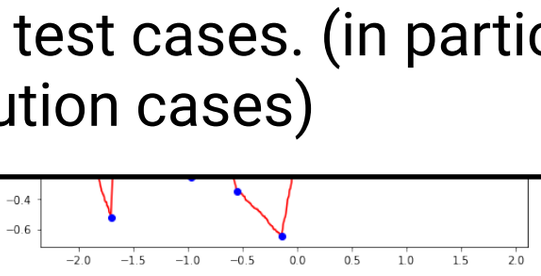


Extra Trees (bootstrap)

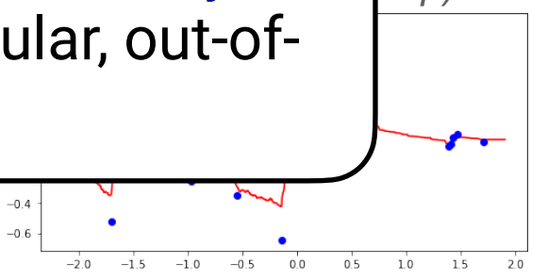


Gradient Boosting





trap)



But we often still see the Rashomon (i.e. similar CV performances)

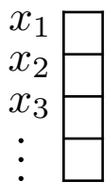
and these can predict *very differently* for further test cases. (in particular, out-of-distribution cases)

Designing relevant “inductive biases”

Use **heuristic assumptions, domain knowledge** to constrain/regularize the model space.

Input representation design,
selection, and its engineering
(descriptors + feature engineering)

vs “Kitchen-sink” models
(feeding every possible descriptor)



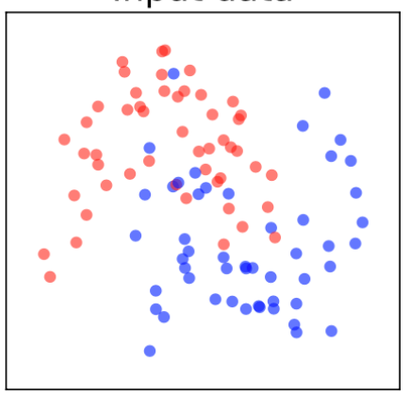
Input variables

Function model

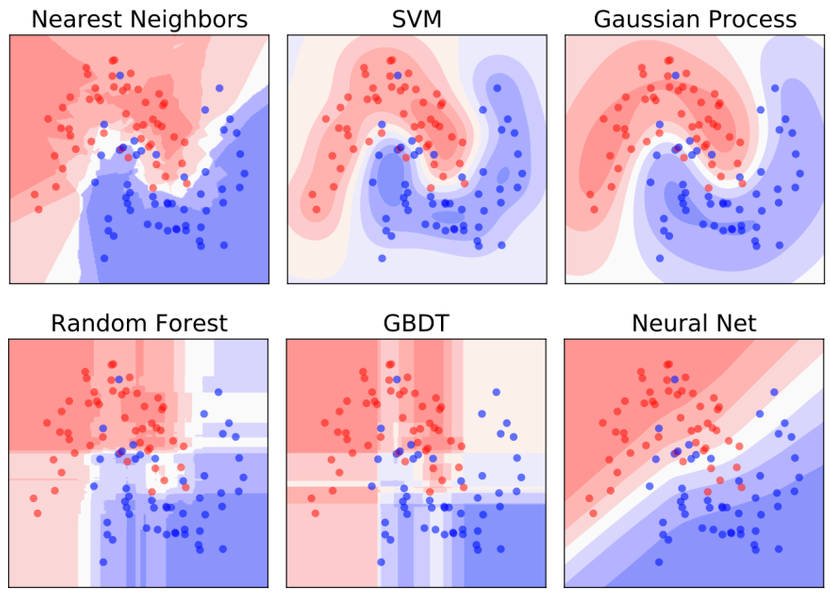
Prediction

Classifier or Regressor

Input data



Every requirement should be explicitly encoded into model, otherwise we need to provide it through examples.
(symmetry, invariance, etc)



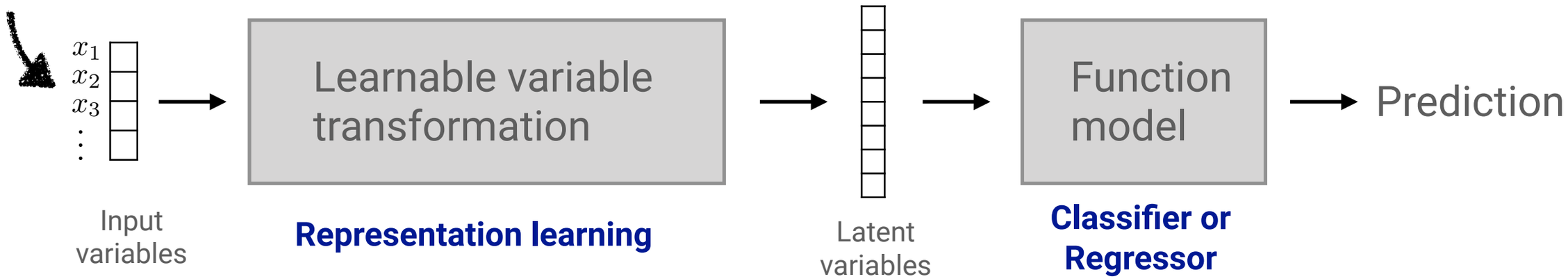
Simple model is enough whenever we can have determining input variables necessary and sufficient to fully define the desired output.

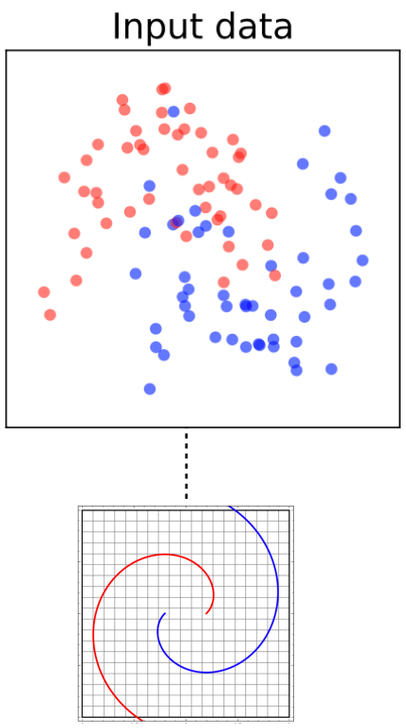
Designing relevant "inductive biases"

Use **heuristic assumptions, domain knowledge** to constrain/regularize the model space.

"Kitchen-sink" (raw) inputs

Architecture design when we go for representation learning



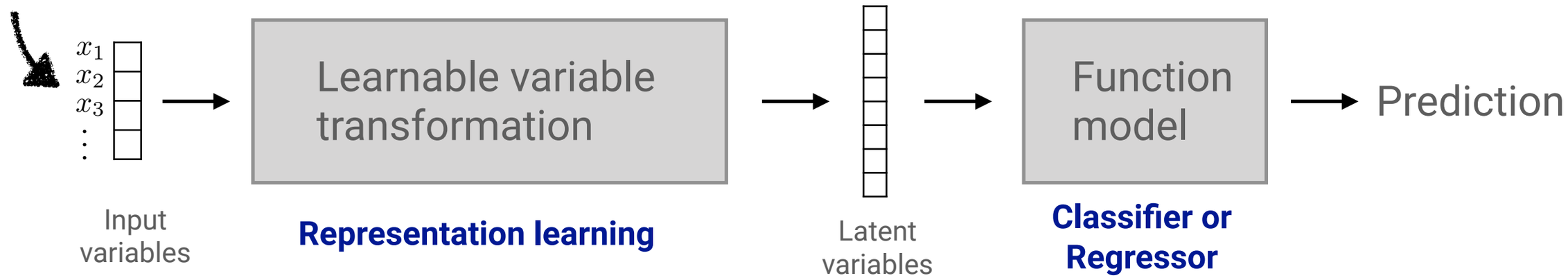


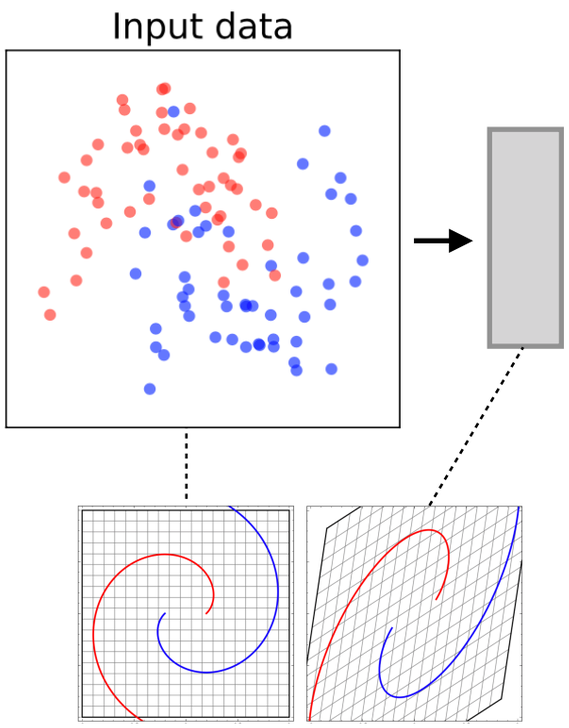
Designing relevant "inductive biases"

Use **heuristic assumptions, domain knowledge** to constrain/regularize the model space.

"Kitchen-sink" (raw) inputs

Architecture design when we go for representation learning



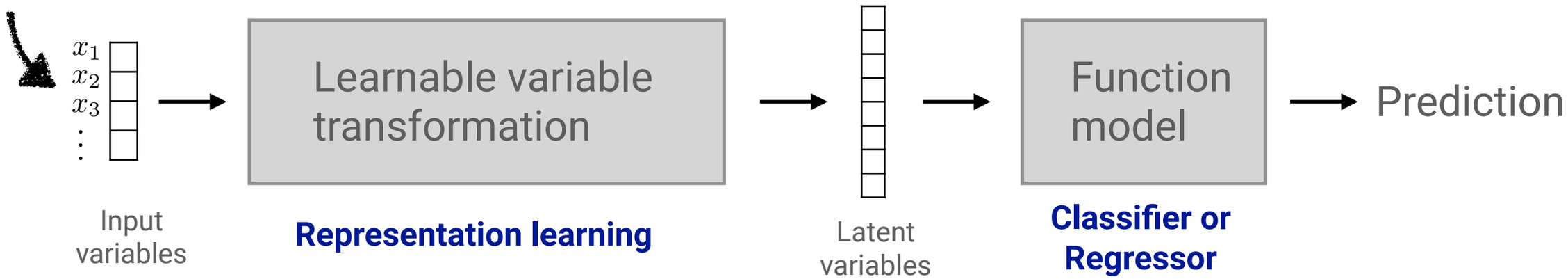


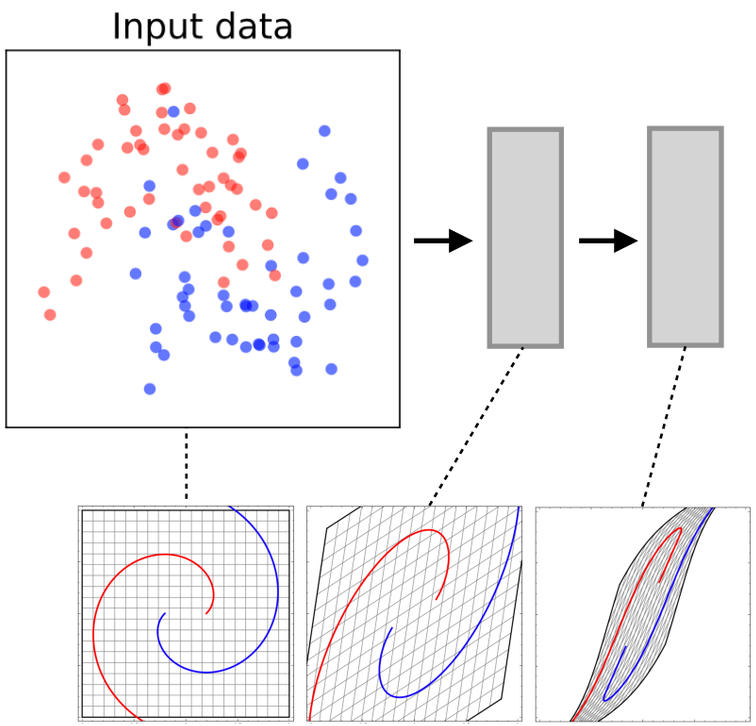
Designing relevant "inductive biases"

Use **heuristic assumptions, domain knowledge** to constrain/regularize the model space.

"Kitchen-sink" (raw) inputs

Architecture design when we go for representation learning



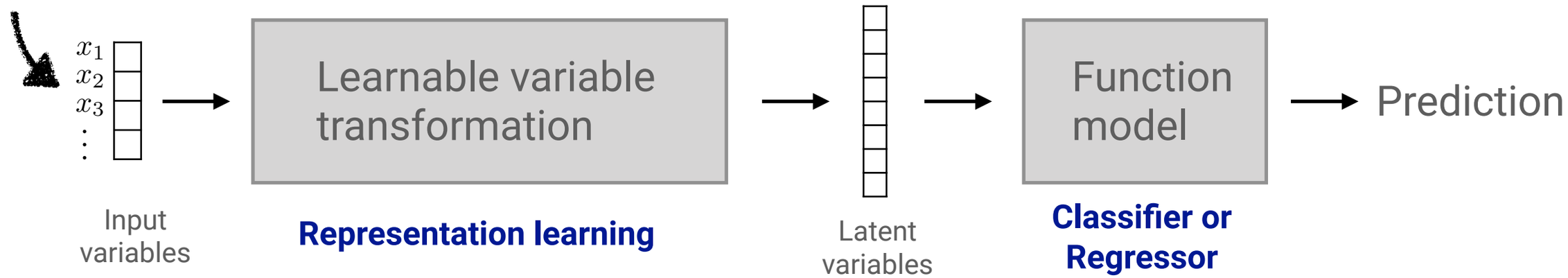


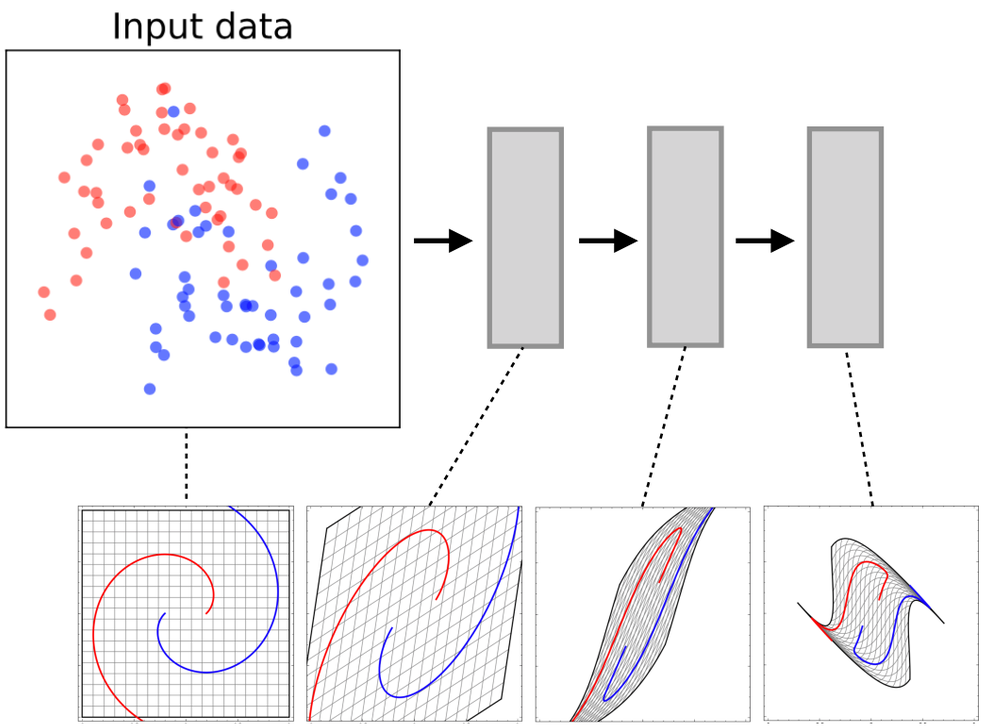
Designing relevant "inductive biases"

Use **heuristic assumptions, domain knowledge** to constrain/regularize the model space.

"Kitchen-sink" (raw) inputs

Architecture design when we go for representation learning



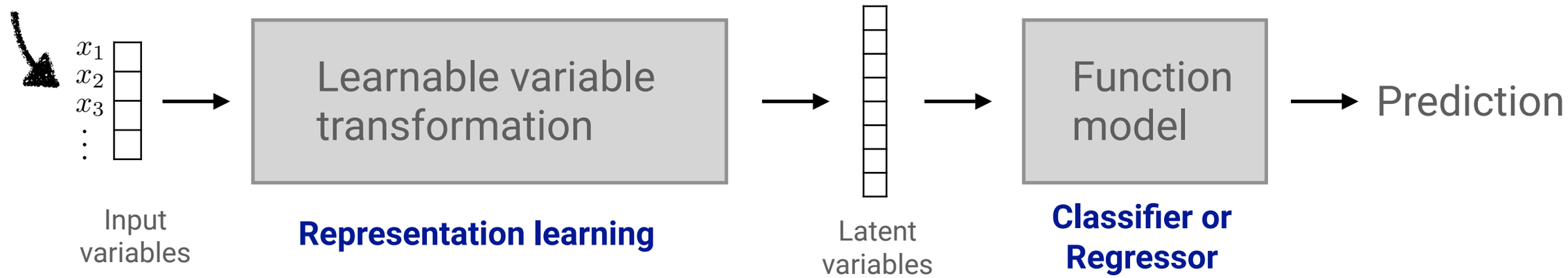


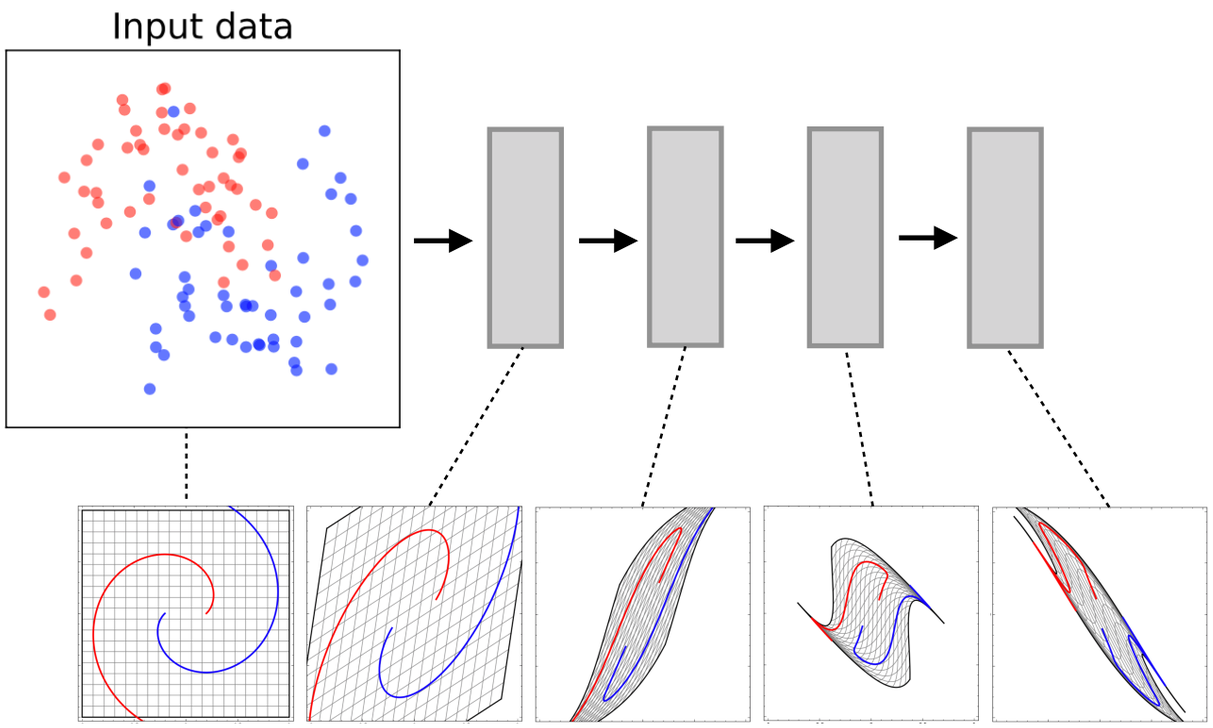
Designing relevant "inductive biases"

Use **heuristic assumptions, domain knowledge** to constrain/regularize the model space.

"Kitchen-sink" (raw) inputs

Architecture design when we go for representation learning



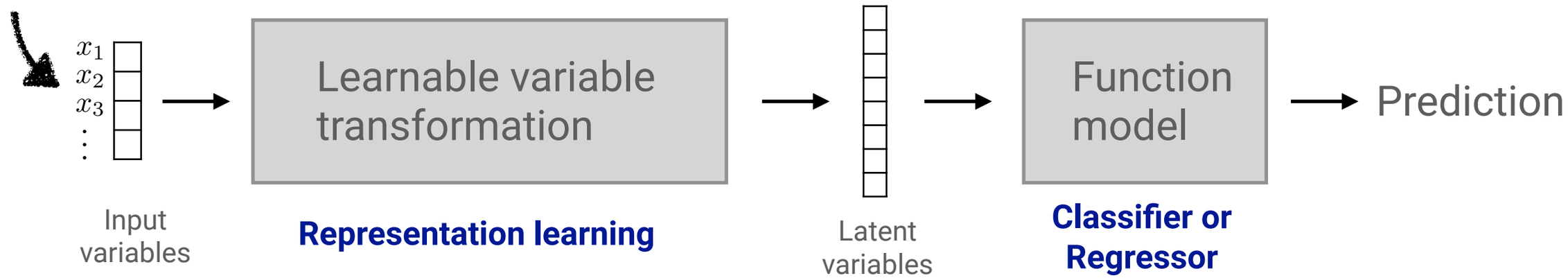


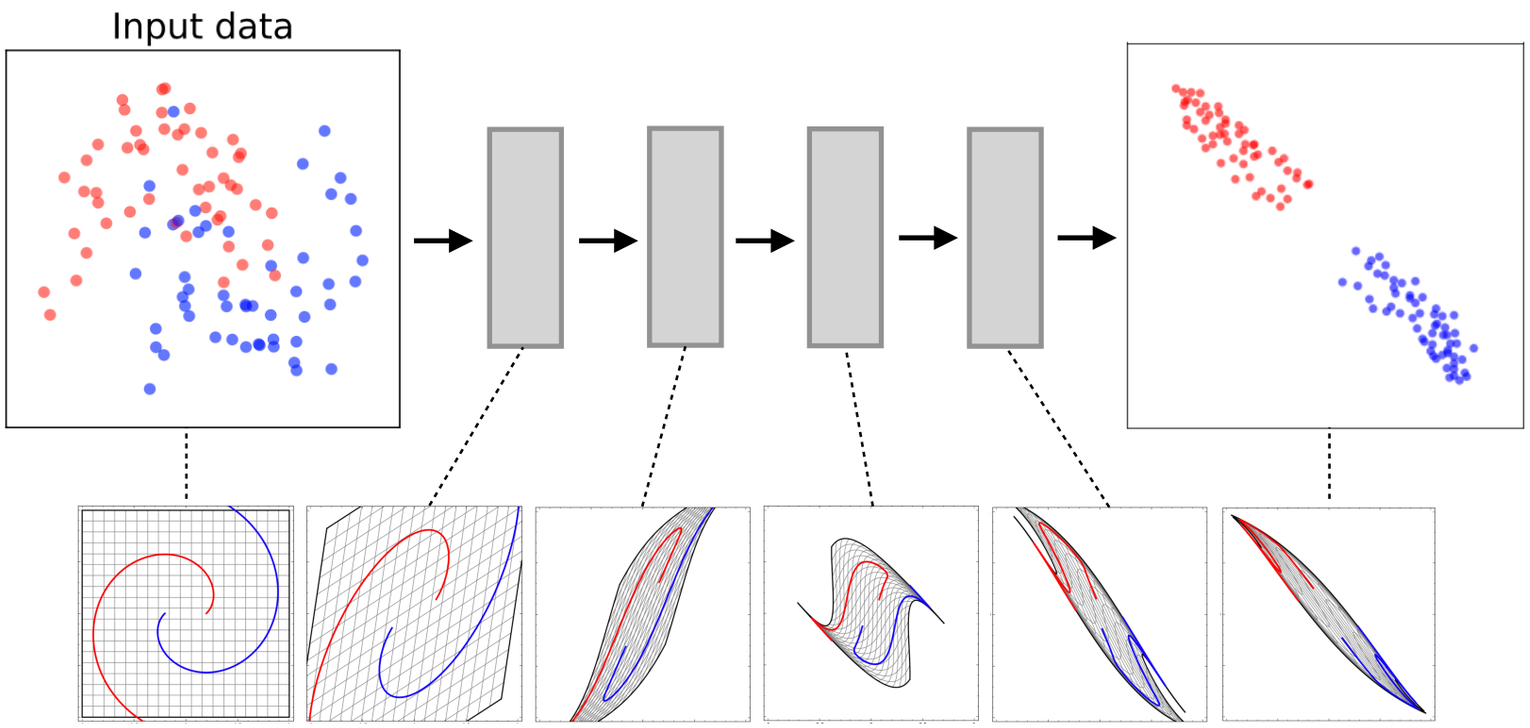
Designing relevant "inductive biases"

Use **heuristic assumptions, domain knowledge** to constrain/regularize the model space.

"Kitchen-sink" (raw) inputs

Architecture design when we go for representation learning



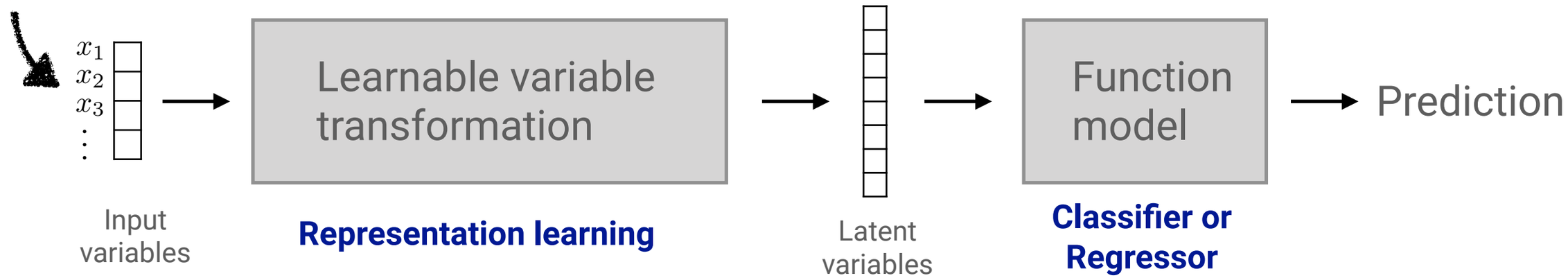


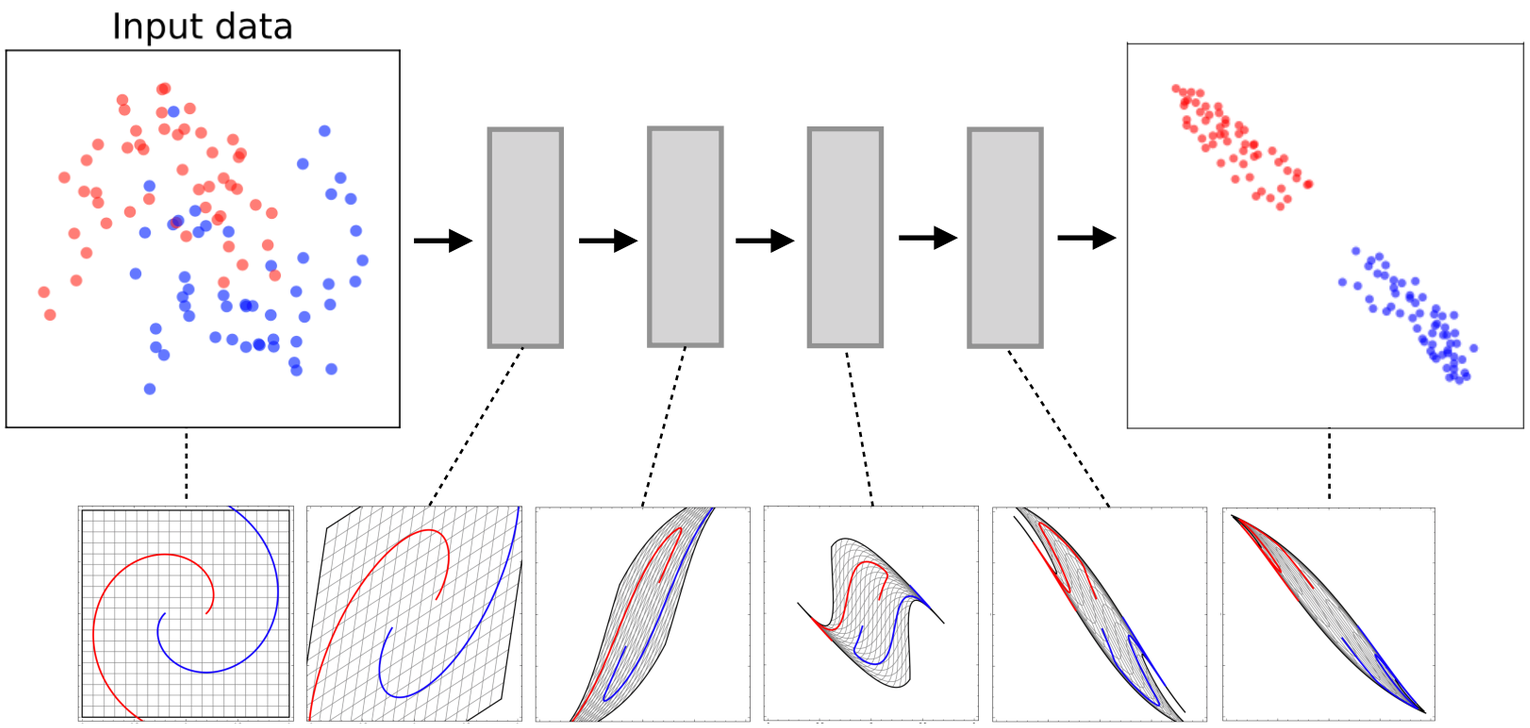
Designing relevant “inductive biases”

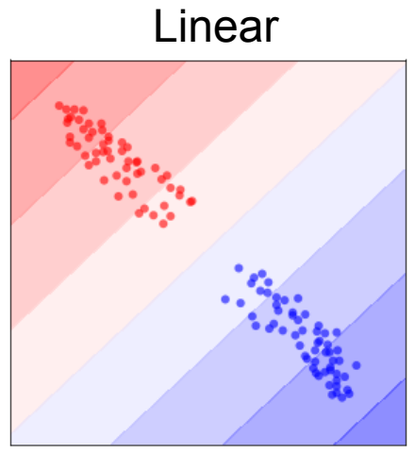
Use **heuristic assumptions, domain knowledge** to constrain/regularize the model space.

“Kitchen-sink” (raw) inputs

Architecture design when we go for representation learning



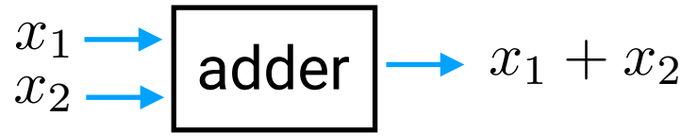




Again, simple model is enough when we have good features.

A toy example: approximate adders

Try to teach ML “arithmetic addition” only by examples.
(the case where a clear answer and underlying logic exist)



train	$1+3=4$	
	$2+5=7$	
	$5+9=14$	
	$3+10=13$	
	$5+6=11$	
test	$1+1=?$	2
	$1+(-1)=?$	0
	$12892+9837=?$	22729

[1]:
```
import numpy as np
```

[2]:
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.neural_network import MLPRegressor
from sklearn.linear_model import LinearRegression
```

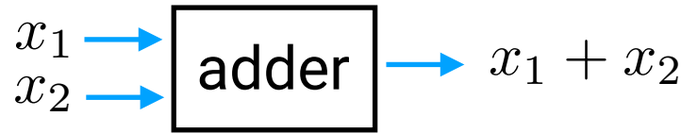
[3]:
```
np.set_printoptions(precision=5, suppress=True)
```

[4]:
```
X = [[1, 3], [2, 5], [5, 9], [3, 10], [5, 6]]
y = [4, 7, 14, 13, 11]
```

We can also add both 5+6 and 6+5 to tell ML
by examples that addition is commutative.

A toy example: approximate adders

Try to teach ML “arithmetic addition” only by examples.
(the case where a clear answer and inductive logic exist)



train	$1 + 3 = 4$	
	$2 + 5 = 7$	
	$5 + 9 = 14$	
	$3 + 10 = 13$	
	$5 + 6 = 11$	
test	$1 + 1 = ?$	2
	$1 + (-1) = ?$	0
	$12892 + 9837 = ?$	22729

[1]:
```
import numpy as np
```

[2]:
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.neural_network import MLPRegressor
from sklearn.linear_model import LinearRegression
```

[3]:
```
np.set_printoptions(precision=5, suppress=True)
```

[4]:
```
X = [[1, 3], [2, 5], [5, 9], [3, 10], [5, 6]]
y = [4, 7, 14, 13, 11]
```

[5]:
```
model = RandomForestRegressor()
model.fit(X, y)
model.predict(X)
```

[5]:
```
array([ 5.15,  7.22, 13.04, 11.79, 10.3 ])
```

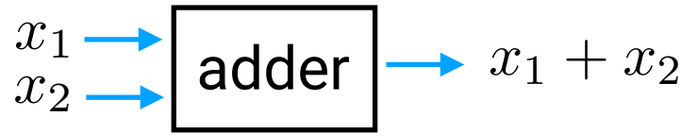
[6]:
```
model.predict([[1, 1], [1, -1], [12892, 9837]])
```

[6]:
```
array([ 5.15,  5.15, 12.75])
```

RF says 1 + 1 = 5.15 and 1 - 1 = 5.15 and 12892 + 9837 = 12.75

A toy example: approximate adders

Try to teach ML “arithmetic addition” only by examples.
(the case where a clear answer and inductive logic exist)



train	$1+3=4$	
	$2+5=7$	
	$5+9=14$	
	$3+10=13$	
	$5+6=11$	
test	$1+1=?$	2
	$1+(-1)=?$	0
	$12892+9837=?$	22729

```
[1]: import numpy as np

[2]: from sklearn.ensemble import RandomForestRegressor
     from sklearn.neural_network import MLPRegressor
     from sklearn.linear_model import LinearRegression

[3]: np.set_printoptions(precision=5, suppress=True)

[4]: X = [[1, 3], [2, 5], [5, 9], [3, 10], [5, 6]]
     y = [4, 7, 14, 13, 11]

[5]: model = RandomForestRegressor()
     model.fit(X, y)
     model.predict(X)

[5]: array([ 5.15,  7.22, 13.04, 11.79, 10.3 ])

[6]: model.predict([[1, 1], [1, -1], [12892, 9837]])

[6]: array([ 5.15,  5.15, 12.75])
```

RF says 1 + 1 = 5.15 and 1 - 1 = 5.15 and 12892 + 9837 = 12.75

```
[7]: model = MLPRegressor(hidden_layer_sizes=(10, 5), activation="relu", max_iter=1000)
     model.fit(X, y)
     model.predict(X)

[7]: array([ 5.86059,  8.1815 , 13.25118, 13.34199, 10.41161])

[8]: model.predict([[1, 1], [1, -1], [12892, 9837]])

[8]: array([    3.96754,     2.51966, 14506.35644])
```

2-10-5-1 Feed Forward NN (MLP, Multi-Layer Perceptron)

MLP better? But anyway it’s totally wrong.

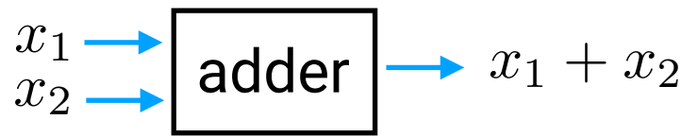
A toy example: approximate adders

Linear regression rocks! 😆

```
[9]: model = LinearRegression()
     model.fit(X, y)
     model.predict(X)

[9]: array([ 4.,  7., 14., 13., 11.])

[10]: model.predict([[1, 1], [1, -1], [12892, 9837]])

[10]: array([    2.,    -0., 22729.])
```

LR says 1 + 1 = 2 and 1 - 1 = 0 and 12892 + 9837 = 22729

(Perfect Answers!!)



train	$1+3=4$	
	$2+5=7$	
	$5+9=14$	
	$3+10=13$	
	$5+6=11$	
test	$1+1=?$	2
	$1+(-1)=?$	0
	$12892+9837=?$	22729

A toy example: approximate adders

Linear regression rocks! 😆

```
[9]: model = LinearRegression()
     model.fit(X, y)
     model.predict(X)
```

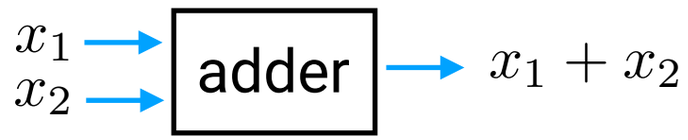
```
[9]: array([ 4.,  7., 14., 13., 11.])
```

```
[10]: model.predict([[1, 1], [1, -1], [12892, 9837]])
```

```
[10]: array([    2.,    -0., 22729.])
```

LR says 1 + 1 = 2 and 1 - 1 = 0 and 12892 + 9837 = 22729

(Perfect Answers!!)



train	$1+3=4$	
	$2+5=7$	
	$5+9=14$	
	$3+10=13$	
	$5+6=11$	
test	$1+1=?$	2
	$1+(-1)=?$	0
	$12892+9837=?$	22729

All are because of **“inductive bias”** intrinsically encoded in the model.

```
model.coef_, model.intercept_
```

```
(array([1., 1.]), -1.7763568394002505e-15)
```

LR with two input variables is just fitting a plane $ax_1 + bx_2 + c$ to points in 3D

Any three instances are enough to have $a = 1, b = 1, c = 0 \Rightarrow x_1 + x_2$

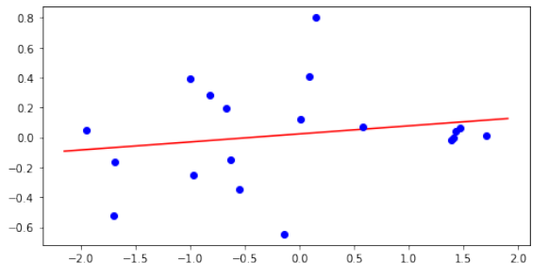
MLP has partial “linearity” inside and that’s why MLP is better than RF is.

RF is “piecewise constant” and only returns values between sample min and max.

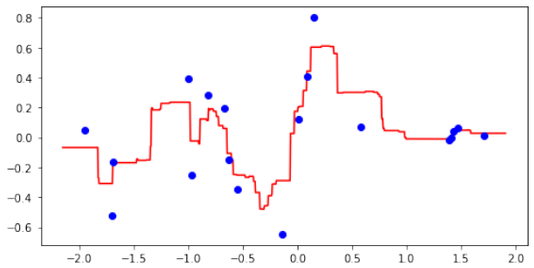
Designing relevant “inductive biases”

“Inductive biases” (often unintendedly) defines **the interspace of instances**, and then also defines the prediction for unseen areas (crucial for out-of-distribution predictions).

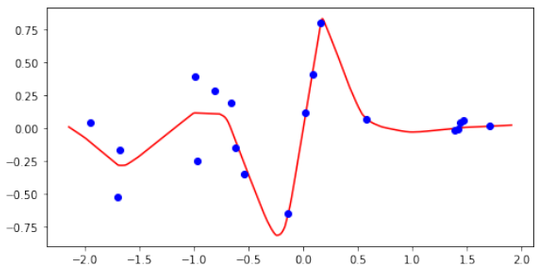
Linear Regression



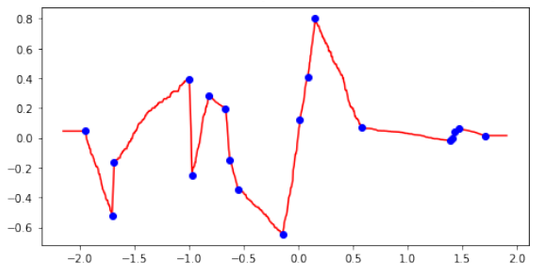
Random Forest



Neural Networks (ReLU)



Extra Trees (no bootstrap)



Small ← Data Required To Make ML Work → **Large**

Simple ← Input-Output Correlation for Target → **Complex**

Conservative models + Strong inductive biases that best fit to the given problem

Carefully designed inputs (input only confidently relevant info into ML)

General models having a large number of parameters + Generalizable inductive biases (enables zero-shot/few-shot transfer?)

Kitchen-sink inputs (input all potentially relevant info into ML)

⇢ Use any “physics-informed” conditions to further constrain or regularize the model space sounds a good idea indeed

Gaps between technical interests and reality

Our technical interests. we're very excited to explore ML over large data (for pretraining + transfer) with generalizable modular structures: *CNNs vs Transformers vs GNNs vs MLPs*

A recent news: OGB Large-Scale Challenge @ KDDCup 2021

PCQM4M-LSC predicting DFT-calculated HOMO-LUMO energy gap of molecules given their 2D molecular graphs. (3,803,453 graphs from PubChemQC; cf. 133,885 graphs for QM9)

OGB OPEN GRAPH BENCHMARK

1st place:	Test MAE 0.1200 (eV)	**10 GNNs (12-Layer Graphormer) + 8 ExpC*s (5-Layer ExpandingConv)**
2nd place:	Test MAE 0.1204 (eV)	**73 GNNs (11-Layer LiteGEMConv with Self-Supervised Pretraining)**
3rd place:	Test MAE 0.1205 (eV)	**20 GNNs (32-Layer GN with Noisy Nodes)**

Current ML is too data-hungry (and purportedly vulnerable to any data bias)

Modern ML can learn any input-output mappings in theory, but more data is needed when the input-output correlation is weak.

Our reality. Practical "open-end" cases
we can only have very limited data relative to the astronomically vast search space.

Today's talk

Our struggles for better ML practices with underspecified, sparse, biased observational data (i.e. a collection of experimental facts from literature)

ChemCatChem
The European Society Journal for Catalysis
Chemistry Europe
European Chemical Societies Publishing

Analysis of Updated Literature Data up to 2019 on the Oxidative Coupling of Methane Using an Extrapolative Machine-Learning Method to Identify Novel Catalysts

Dr. Shinya Mine, Motoshi Takao, Taichi Yamaguchi, Dr. Takashi Toyao ✉, Dr. Zen Maeno, Dr. S. M. A. Hakim Siddiki, Dr. Satoru Takakusagi, Prof. Ken-ichi Shimizu ✉, Dr. Ichigaku Takigawa ✉

First published: 31 May 2021 | https://doi.org/10.1002/cctc.202100495

ChemCatChem
The European Society Journal for Catalysis
Chemistry Europe
European Chemical Societies Publishing

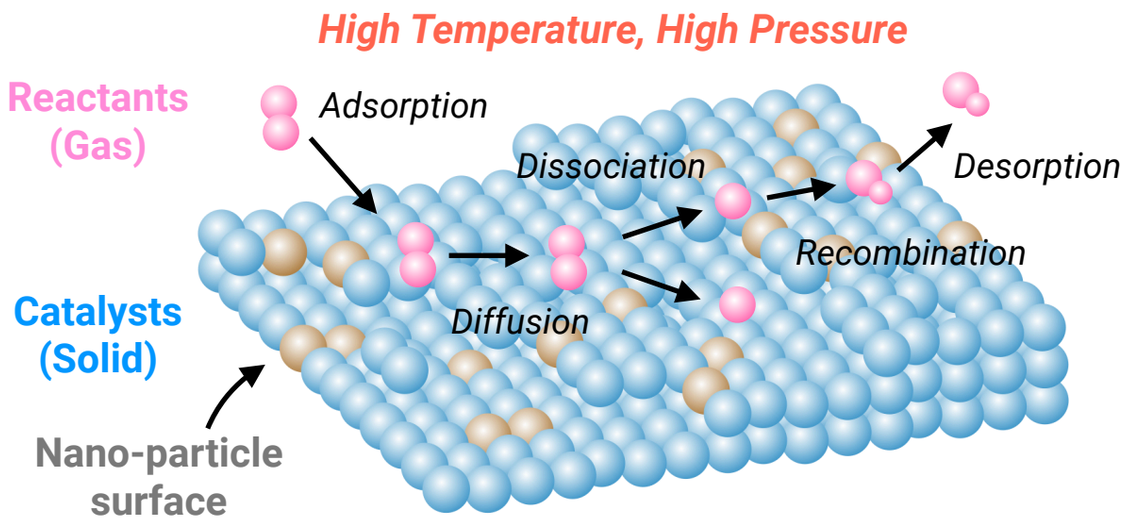
Statistical Analysis and Discovery of Heterogeneous Catalysts Based on Machine Learning from Diverse Published Data

Keisuke Suzuki, Dr. Takashi Toyao, Dr. Zen Maeno, Dr. Satoru Takakusagi, Prof. Ken-ichi Shimizu ✉, Dr. Ichigaku Takigawa ✉

First published: 09 July 2019 | https://doi.org/10.1002/cctc.201900971 | Citations: 10

Gas-phase reactions on sold-phase catalyst surface (Heterogeneous catalysis)

Industrial Synthesis (e.g. Haber-Bosch), Automobile Exhaust Gas Purification, Methane Conversion, etc.



Devilishly complex too-many-factor process!!

God made the bulk;
the **surface** was invented by the **devil**

—— Wolfgang Pauli

The base dataset

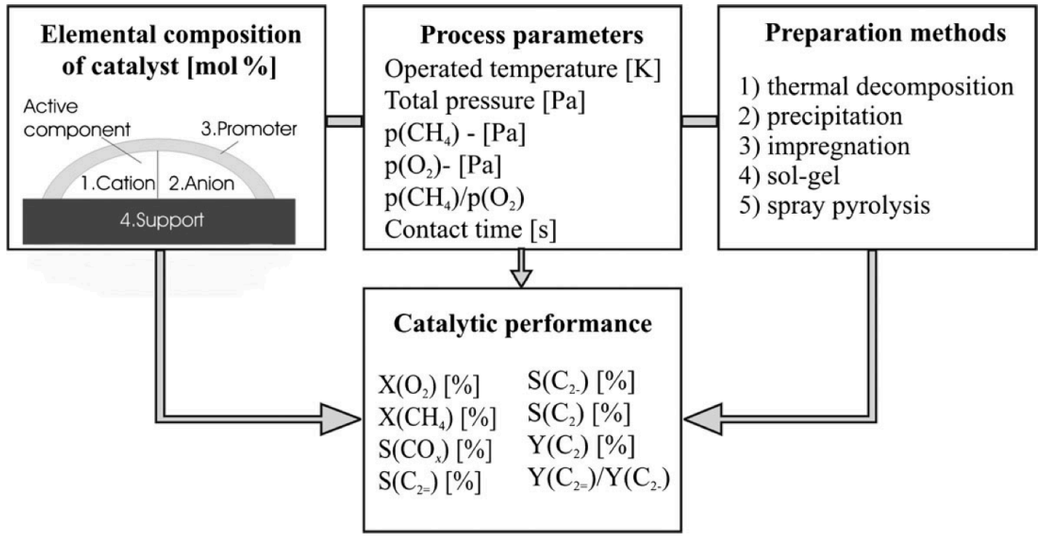
1866 catalyst records from 421 reports

Oxidative coupling of methane (OCM) reactions

Methane (CH_4) is partially oxidized to **C_2 hydrocarbons** such as ethane (C_2H_6) and ethylene (C_2H_4) in a single step

- Zavyalova, U.; Holena, M.; Schlögl, R.; Baerns, *ChemCatChem* 2011.
- Followup:
  Kondratenko, E. V.; Schlüter, M.; Baerns, M.; Linke, D.; Holena, M. *Catal. Sci. Technol.* 2015.
- Renalysis with Corrections & Outlier Removal
  Schmack, R.; Friedrich, A.; Kondratenko, E. V.; Polte, J.; Werwatz, A.; Kraehnert, R. *Nat Commun* 2019.

http://www.fhi-berlin.mpg.de/acnew/department/pages/ocmdata.html
https://www.nature.com/articles/s41467-019-08325-8#Sec19



Elemental composition of catalyst (mol%) | Process parameters + Preparation | Catalytic performance

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG
1	Nr of publication	Cation 1		Cation 2		Cation 3		Cation 4		Anion 1		Anion 2		Pro-motor	Support 1		Support 2		Preparation	Temperature, K	p(CH_4), bar	p(O_2), bar	p(CH_4)/p(O_2)	P total, bar	Contact time, s	$X(O_2)$, %	$X(CH_4)$, %	$S(CO_x)$, %	$S(C_2^=)$, %	$S(C_2^-)$, %	$S(C_2)$, %	$Y(C_2)$,%
2		value	mol%	value	mol%	value	mol%	value		value		value			value	mol%	value	mol%														
212		Li	22.2	Ca	77.8														n.a.	984	0.47	0.11	4.4	1.0	8.70		18.0	30.0	38.0	27.0	65.0	11.7
213		Na	7.3	Ca	92.7														n.a.	982	0.47	0.11	4.4	1.0	8.70		16.0	25.0	35.0	35.0	70.0	11.2
214		Li	17.2	Mg	82.8														n.a.	1043	0.38	0.13	3.0	1.0	5.50		59.0	48.0	36.0	17.0	53.0	31.3
215		Sn	1.2	Li	17.6	Mg	81.20												n.a.	977	0.47	0.11	4.4	1.0	7.50		23.0	31.0	35.0	27.0	62.0	14.3
216	23	Li	22.2	Ca	77.8														n.a.	984	0.47	0.11	4.4	1.0	8.70		18.0	30.0	38.0	27.0	65.0	11.7
217		Na	7.3	Ca	92.7														n.a.	982	0.47	0.11	4.4	1.0	8.70		16.0	25.0	35.0	35.0	70.0	11.2
218		Na	8.5	Pb	8.5	Si	8.50	Zn	74.50										n.a.	1023	0.90	0.10	9.0	1.0	2.40	71.0	8.0		34.0	34.0	68.0	5.4
219		Na	8.0	K	4.0	Co	20.00	Zr	50.00	S	16.0	P	2.0	Cl					Therm.decomp.	970	0.50	0.11	4.8	1.0	1.20		19.2		43.2	17.6	60.8	11.7
220		Zr	100.0																n.a.	973	0.47	0.11	4.2	1.0	1.20		12.0		0.0	2.5	2.5	0.3
221		Na	8.5	Pb	8.5	Si	8.50	Zn	74.50										n.a.	1023	0.90	0.10	9.0	1.0	2.40	71.0	8.0		34.0	34.0	68.0	5.4
222		Cs	100.0											Cl					n.a.	1023	0.91	0.09	10.0	1.0	2.00		5.0		30.0	54.0	84.0	4.2
223		Li	100.0											Cl					n.a.	1023	0.91	0.09	10.0	1.0	2.00		5.0		44.0	46.0	90.0	4.5
224		Li	33.3	Ca	33.3	Pr	33.40							Cl					n.a.	1023	0.91	0.09	10.0	1.0	2.00		14.0		71.0	13.0	84.0	11.8
225		Li	33.3	Ca	33.3	Ce	33.40							Cl					n.a.	1023	0.91	0.09	10.0	1.0	2.00		14.0		70.0	14.0	84.0	11.8
226		Ca	100.0											Cl					n.a.	1023	0.67	0.07	10.0	1.0	2.00		14.0		71.0	11.0	82.0	11.5
227		Mg	100.0											Cl					n.a.	1023	0.67	0.07	10.0	1.0	2.00		5.0		38.0	49.0	87.0	4.4
228		Mn	100.0											Cl					n.a.	1023	0.67	0.07	10.0	1.0	2.00		14.0		71.0	13.0	84.0	11.8
229		Pb	100.0											Cl					n.a.	1023	0.67	0.07	10.0	1.0	2.00		10.0		48.0	18.0	66.0	6.6

Problem #1: It's underspecified

- Each catalyst was mostly measured at different reaction conditions.
- Only a few are measured under multiple conditions

158 compositions > 2 conditions

60 compositions > 3 conditions

26 compositions > 4 conditions

La:100.0 x 24
Mg:100.0 x 18
Ca:100.0 x 18
Sm:100.0 x 13
Nd:100.0 x 10
Ba:100.0 x 9
Y:100.0 x 9
Ce:100.0 x 9
Zr:100.0 x 9
Sr:100.0 x 7
Si:100.0 x 7
Gd:100.0 x 7
Na:8.9 Si:83.1 Mn:3.5 W:4.5 x 7
Pr:100.0 x 6
Eu:100.0 x 6
Yb:100.0 x 5
Al:100.0 x 4
Li:10.0 Mg:90.0 x 4
Mg:90.9 La:9.1 x 4
Li:9.1 Mg:90.9 x 4
Tb:100.0 x 4
Li:20.0 Cl:20.0 Zr:60.0 x 4
Na:20.0 Cl:20.0 Zr:60.0 x 4
Cl:20.0 K:20.0 Zr:60.0 x 4
Cl:20.0 Rb:20.0 Zr:60.0 x 4
Cl:20.0 Zr:60.0 Cs:20.0 x 4

- No replicates in the same conditions
- But as we see later, reaction conditions are quite influential. Because of this, "no generally valid correlation between a catalyst's composition, its structure and its OCM performance has been established yet."

Observational study ⟷ **Interventional study**

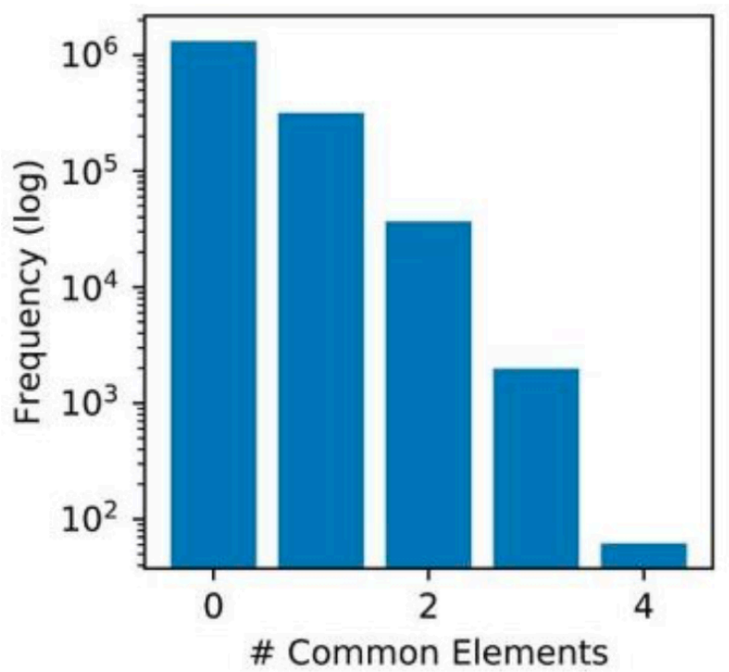
Strong limitation of observational data (just passively acquired)

Problem #2: It's sparse

74 elements

Li	Be	B	C	N	F	Na	Mg	Al	Si	...	Ta	W	Re	Os	Ir	Pt	Au	Tl	Pb	Bi
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	90.800003	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	95.300003	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	95.500000	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	89.599998	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	97.199997	0.0	...	0.0	0.0	0.0	0.0	0.0	2.8	0.0	0.0	0.0	0.000000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
0.0	0.0	0.0	0.0	0.0	0.0	0.0	23.799999	0.000000	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	23.799999
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000

All pairwise comparisons



Mostly, arbitrary pairs of catalysts don't even have any common elements.
Can we meaningfully compare 'Na:33.2 Ti:0.5 Mn:66.3' and 'Zn:77.8 Ce:22.2' ...?

```
for _ in range(20):
    print(random.sample(catalysts, 2))
```

['Na:33.2 Ti:0.5 Mn:66.3', 'Zn:77.8 Ce:22.2']
['C:32.7 K:65.4 Pb:1.9', 'Y:100.0']
['Na:66.7 Mo:33.3', 'Al:94.5 Mo:5.5']
['C:4.0 Na:4.0 Ce:92.0', 'Y:70.0 Bi:30.0']
['Si:98.2 Cs:1.8', 'Ti:50.0 Gd:50.0']
['Na:9.1 Si:82.8 Cr:3.6 W:4.5', 'Na:20.0 Mg:80.0']
['Y:66.7 Ba:33.3', 'Al:77.0 Ag:18.0 Ba:5.0']
['Al:87.0 Cl:8.0 Fe:1.0 Sr:4.0', 'Na:33.2 Mn:66.3 Ta:0.5']
['Na:1.0 La:99.0', 'Li:9.1 Ca:90.9']
['Fe:100.0', 'Sr:50.0 Nd:50.0']

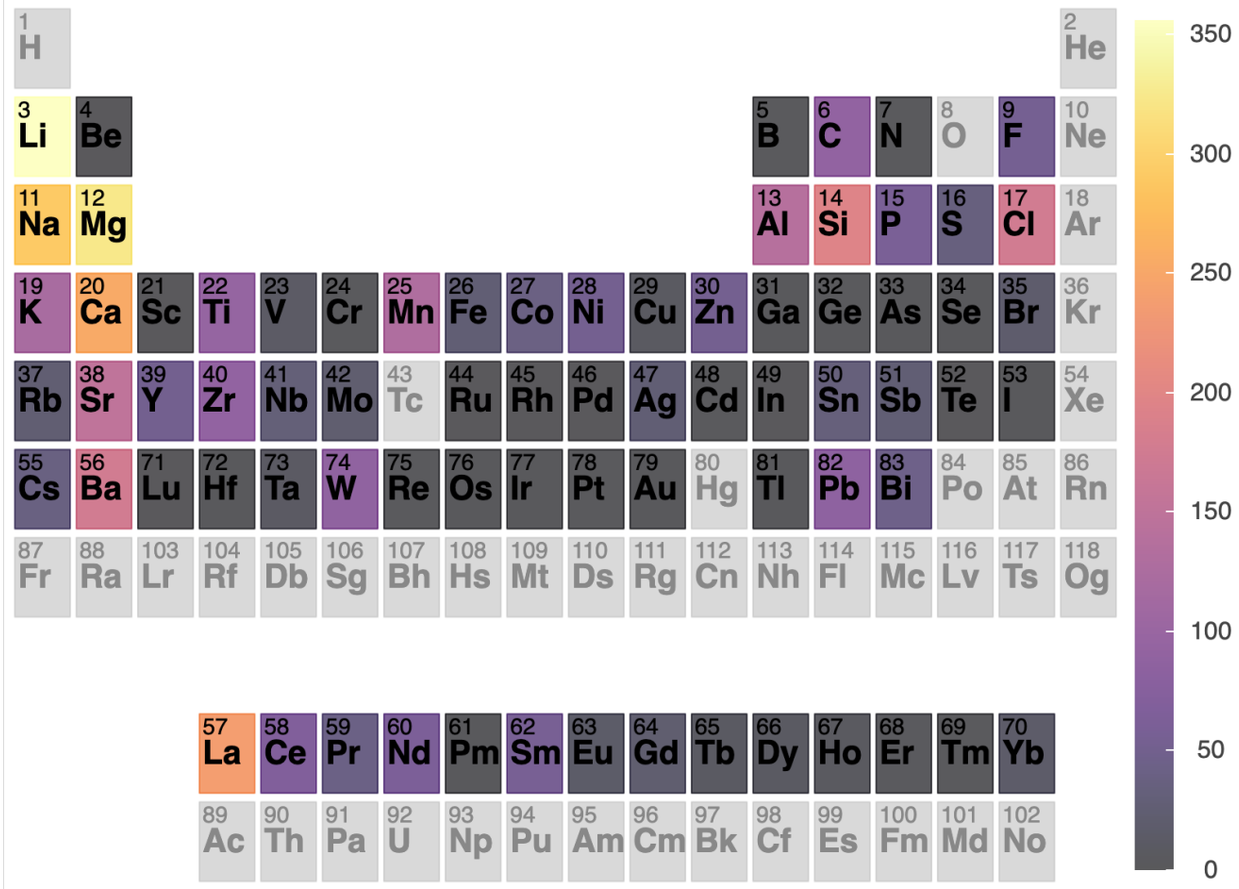
['Al:75.0 Cl:16.0 Sr:8.0 Rh:1.0', 'Na:4.5 Si:79.2 Mn:16.3']
['S:2.9 K:5.7 Ca:91.4', 'Sm:100.0']
['P:34.5 Sr:65.5', 'Na:58.3 Cl:25.0 Mo:16.7']
['Li:23.0 Si:73.2 W:3.8', 'Si:33.3 Ca:66.6 Pb:0.1']
['Al:90.5 Ag:8.5 Pr:1.0', 'Na:66.7 Mo:33.3']
['Cl:20.0 Ba:10.0 Nd:70.0', 'Mg:90.9 La:9.1']
['Gd:100.0', 'Mn:50.0 Mo:50.0']
['Na:76.9 Nb:23.1', 'La:90.0 Pb:10.0']
['Li:6.5 S:3.2 Ca:90.3', 'P:34.5 Sr:65.5']
['Na:5.0 Si:72.0 Cl:5.0 Mn:18.0', 'P:34.0 S:7.5 Ca:51.0 Pb:7.5']

Problem #3: It's biased

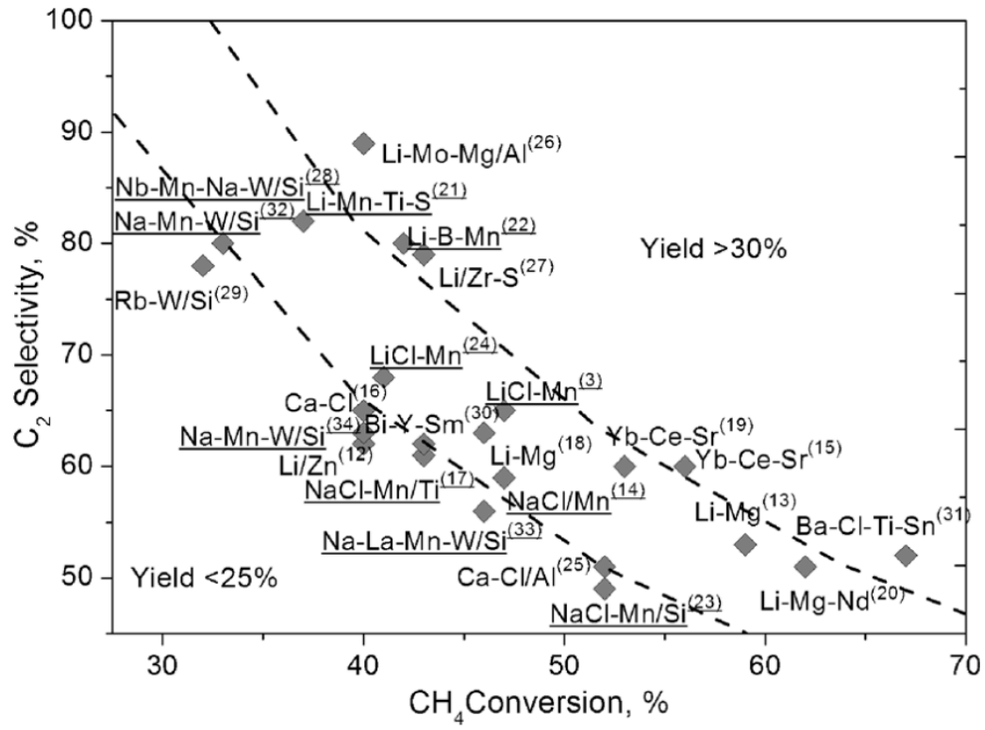
Unavoidable Human-Caused Biases

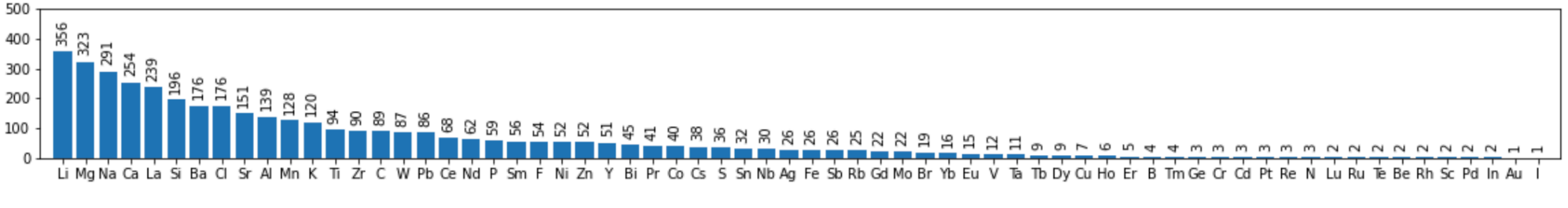
"most chemical experiments are planned by human scientists and therefore are subject to a variety of human cognitive biases, heuristics and social influences."

Jia, X.; Lynch, A.; Huang, Y.; Danielson, M.; Lang'at, I.; Milder, A.; Ruby, A. E.; Wang, H.; Friedler, S. A.; Norquist, A. J.; Schrier, J. *Nature* 2019, 573 (7773), 251–255.



Catalyst such as LaO_3, Li/MgO, and $Mn/Na_2WO_4/SiO_2$ extensively studied.





Our solutions

Solution to Problem #1 (Underspecification)
Tree ensemble regressors with prediction variances are used to make robust and less risky prediction as well as to quantify how uncertain each ML prediction is.

Solution to Problem #2 (Sparsity)
The catalyst representation called SWED (Sorted Weighted Elemental Descriptors) is developed to represent catalysts not in a one-hot fashion but by elemental descriptors.

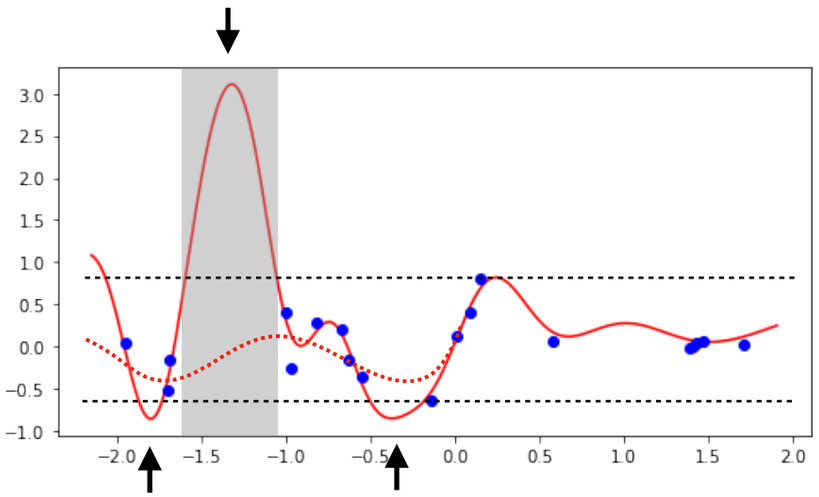
Solution to Problem #3 (Strong Bias)
On the top of the above two, sequential model-based optimization with SWED only by 3 descriptors (electronegativity, density, and ΔH_{fus}) as well as 8 descriptors are explored. Also, for suggested candidates to be worth testing, SHAP interpretations are provided.

OCM Dataset Update, Reanalysis, Exploration
The original dataset (1866 catalyst records from 421 reports until 2009) is extended to 4559 catalyst records from 542 reports from 2010 to 2019, and reanalyzed.

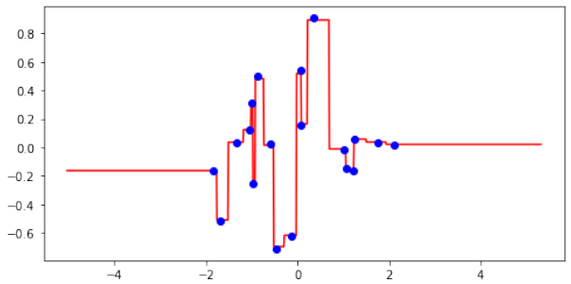
#1. Tree ensemble regression with uncertainty

Tree ensemble regressors with prediction variances are used to make robust and less risky prediction as well as to quantify how uncertain each ML prediction is.

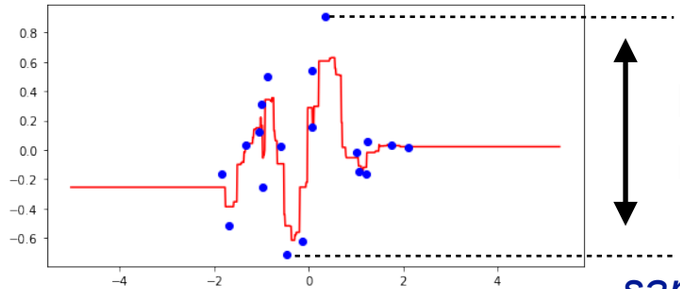
Avoid the risk of unintended extrapolation? (High-dimensional feature spaces can be counterintuitive…)



Gradient Boosted Trees



Random Forest

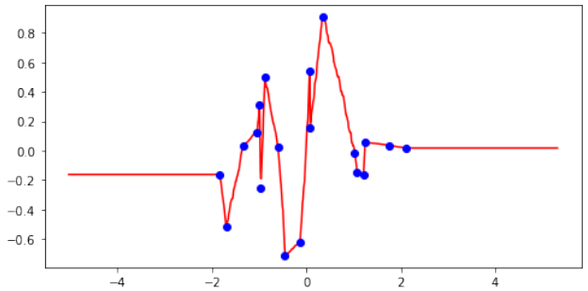


sample max

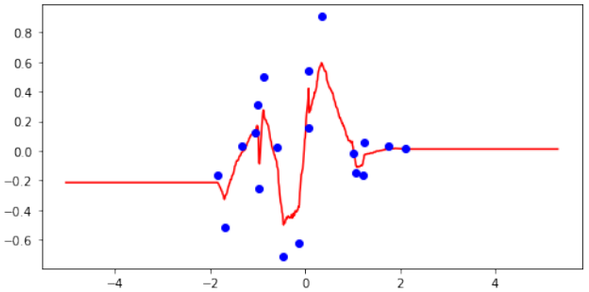
bounded prediction

sample min

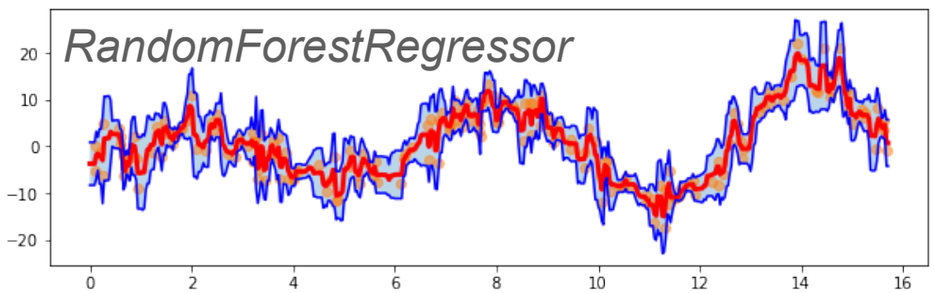
Extra Trees (no bootstrap)

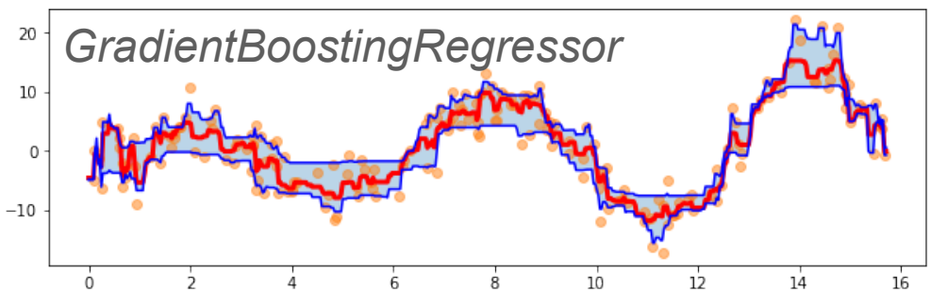


Extra Trees (bootstrap)

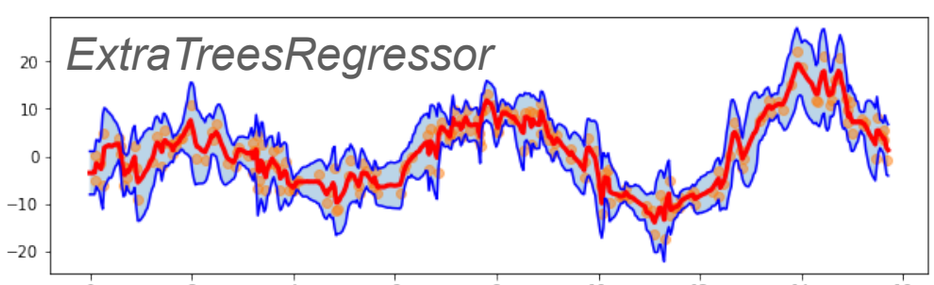


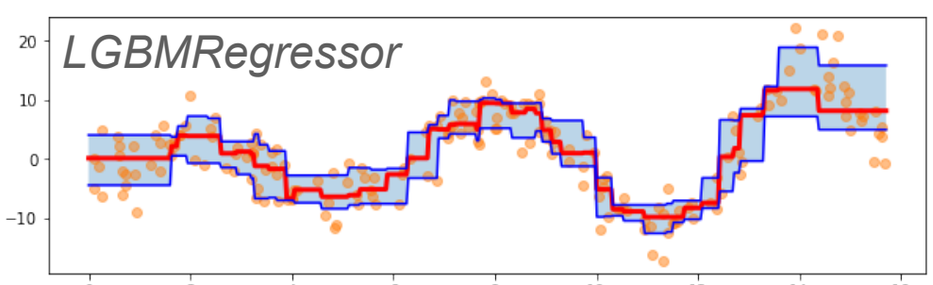
Naturally by the law of total variance





By quantile regression to .16, .5, .84 quantiles





#2. SWED representation of catalysts

The catalyst representation called SWED (Sorted Weighted Elemental Descriptors) is developed to represent catalysts not in a one-hot fashion but by elemental descriptors.

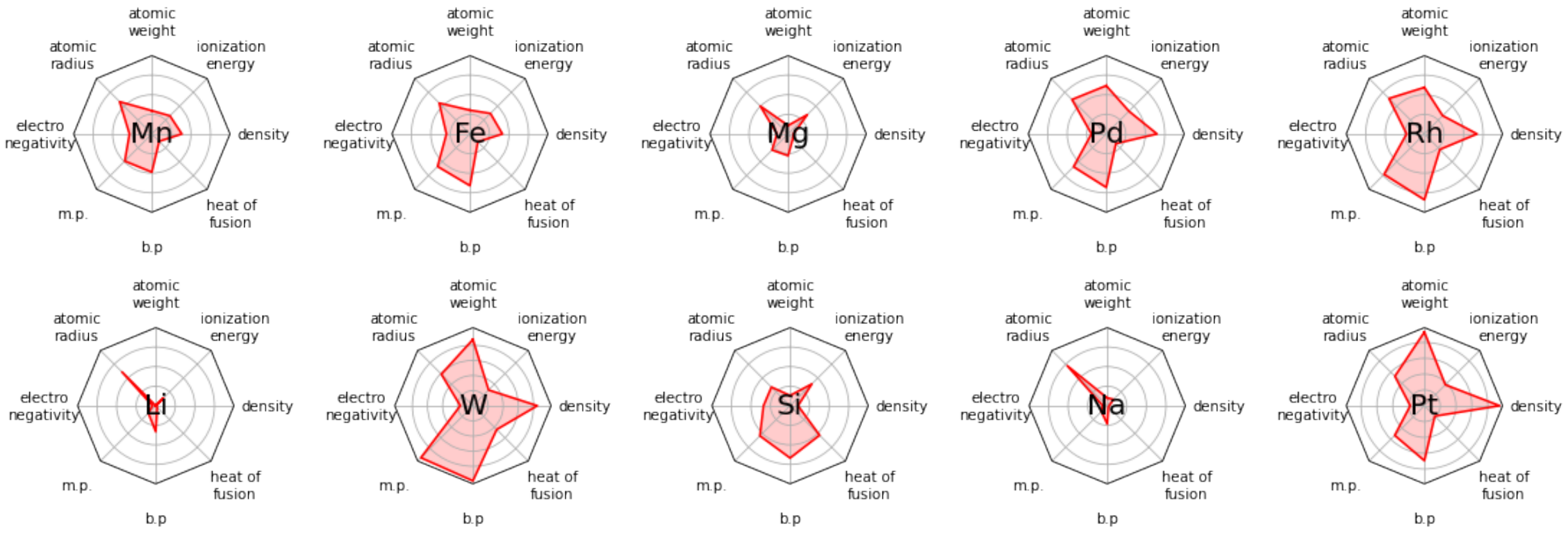
Key Idea

one-hot-like features below are statistically incomparable

Li	Be	B	C	N	F	Na	Mg	Al	Si	...	Ta	W	Re	Os	Ir	Pt	Au	Tl	Pb	Bi
0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

so represent catalysts instead by any elemental descriptors to represent arbitrary (used/unused) elements in a common ground, considering chemical similarities.

Element	Descriptors			
	1	2	...	p
A	A_1	A_2	…	A_p
B	B_1	B_2	…	B_p
C	C_1	C_2	…	C_p
D	D_1	D_2	…	D_p
E	E_1	E_2	…	E_p



SWED (Sorted Weighted Elemental Descriptors)

Literature data

Catalyst	Composition [mol%]				
	A	B	C	D	E
Cat-ABC1	**90**	**4**	**6**	0	0
Cat-BDE1	0	**11**	0	**80**	**9**
Cat-AE1	**75**	0	0	0	**25**
Cat-AE2	**80**	0	0	0	**20**
Cat-ABCDE1	**2**	**3**	**15**	**10**	**70**

Element	Descriptors			
	1	2	…	p
A	A_1	A_2	…	A_p
B	B_1	B_2	…	B_p
C	C_1	C_2	…	C_p
D	D_1	D_2	…	D_p
E	E_1	E_2	…	E_p

Descriptors	8 descriptors (p=8)	3 descriptors (p=3)
AW	○	
Atomic radius	○	
Electronegativity	○	○
m.p.	○	
b.p.	○	
ΔH_{fus}	○	○
Density	○	○
Ionization energy	○	

Sorted Weighted Elemental Descriptor (SWED) representation

K is the max number of elements in a catalyst. K = 5 in this example, K = 8 used in this paper

Catalyst	1st feature				2nd feature				…	Kth feature			
	1	2	…	p	1	2	…	p	…	1	2	…	p
Cat-ABC1	$90\times A_1$	$90\times A_2$	…	$90\times A_p$	$6\times C_1$	$6\times C_2$	…	$6\times C_p$	…	$0\times E_1$	$0\times E_2$	…	$0\times E_p$
Cat-BDE1	$80\times D_1$	$80\times D_2$	…	$80\times D_p$	$11\times B_1$	$11\times B_2$	…	$11\times B_p$	…	$0\times C_1$	$0\times C_2$	…	$0\times C_p$
Cat-AE1	$75\times A_1$	$75\times A_2$	…	$75\times A_p$	$25\times E_1$	$25\times E_2$	…	$25\times E_p$	…	$0\times D_1$	$0\times D_2$	…	$0\times D_p$
Cat-AE2	$80\times A_1$	$80\times A_2$	…	$80\times A_p$	$20\times E_1$	$20\times E_2$	…	$20\times E_p$	…	$0\times D_1$	$0\times D_2$	…	$0\times D_p$
Cat-ABCDE1	$90\times E_1$	$90\times E_2$	…	$90\times E_p$	$15\times C_1$	$15\times C_2$	…	$15\times C_p$	…	$2\times A_1$	$2\times A_2$	…	$2\times A_p$

+

Experimental condition				
Imp.	SG	Pre	Temp.	P_{CH4}/P_{O2}
1	0	0	1023	3
0	1	0	1023	2.5
0	0	0	923	3
0	0	1	923	5
1	0	0	973	4

- Product terms can represent interaction effects between variables (e.g. probabilistic gating, attention, …) and furthermore, they can zero out the feature when the corresponding element is 0%.
- Sorted concatenation is lossless, and was better than weighted sum or weighted max.
- SWED lose the exact composition. To compensate, we also developed a **SWED→composition estimator**.
- We tried many other things (matrix decomposition, Aitchison geometry, GNN, etc) that didn't work.

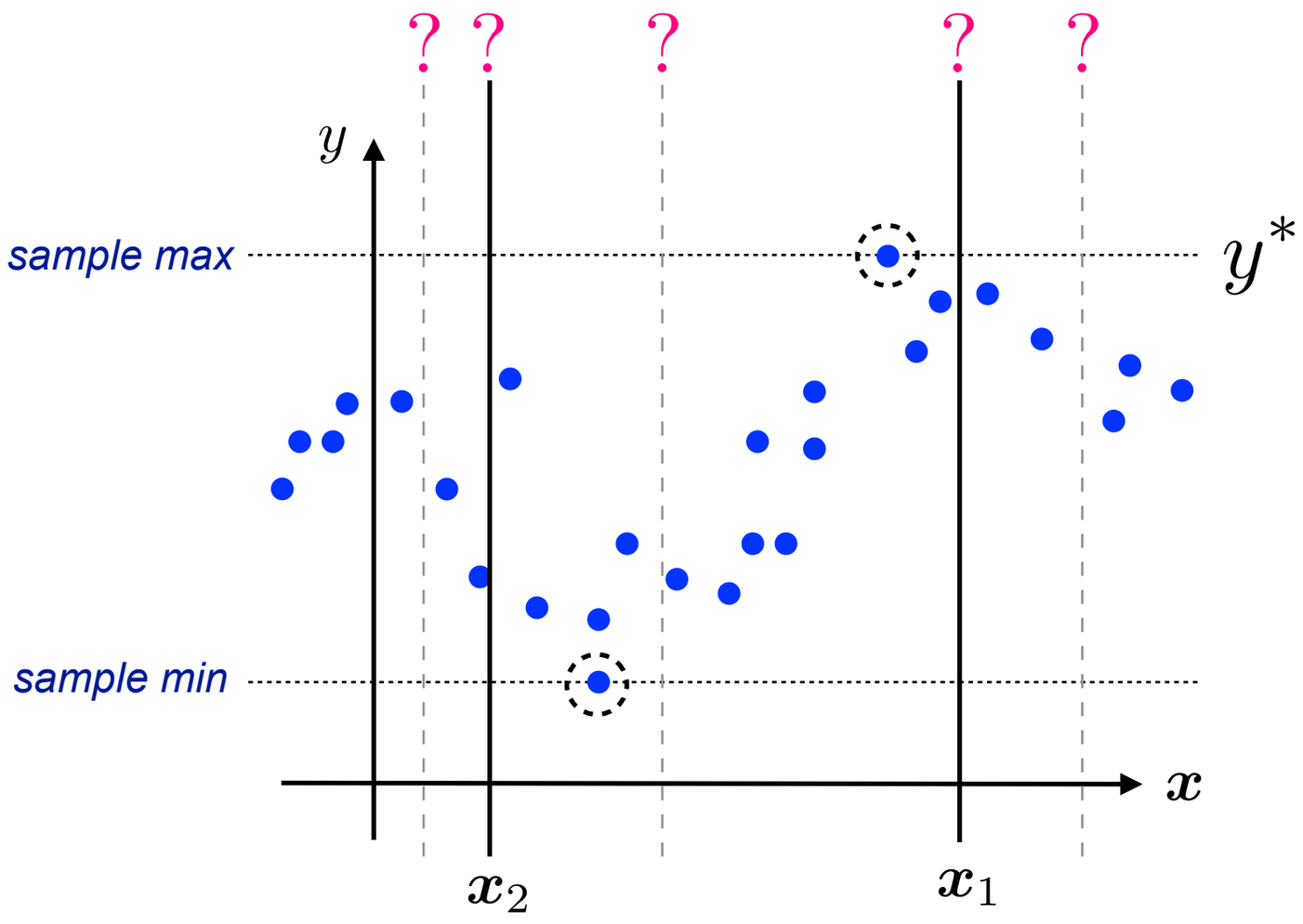
#3. Optimism in the face of uncertainty

We would like to find X better than (hopefully) **the currently known best** y^*.

What next location of $\boldsymbol{x}$ is likely to give higher y?

A fundamental choice: exploitation-exploration tradeoff

Exploitation Make the best decision given current information
Exploration Gather more information by probing uncertain areas



$\boldsymbol{x}_1$ exploitative choice

$\boldsymbol{x}_2$ explorative choice

Random choice (e.g. random design) or evenly spaced sampling (e.g. full factorial design) can also work for lower dimensional exploration.

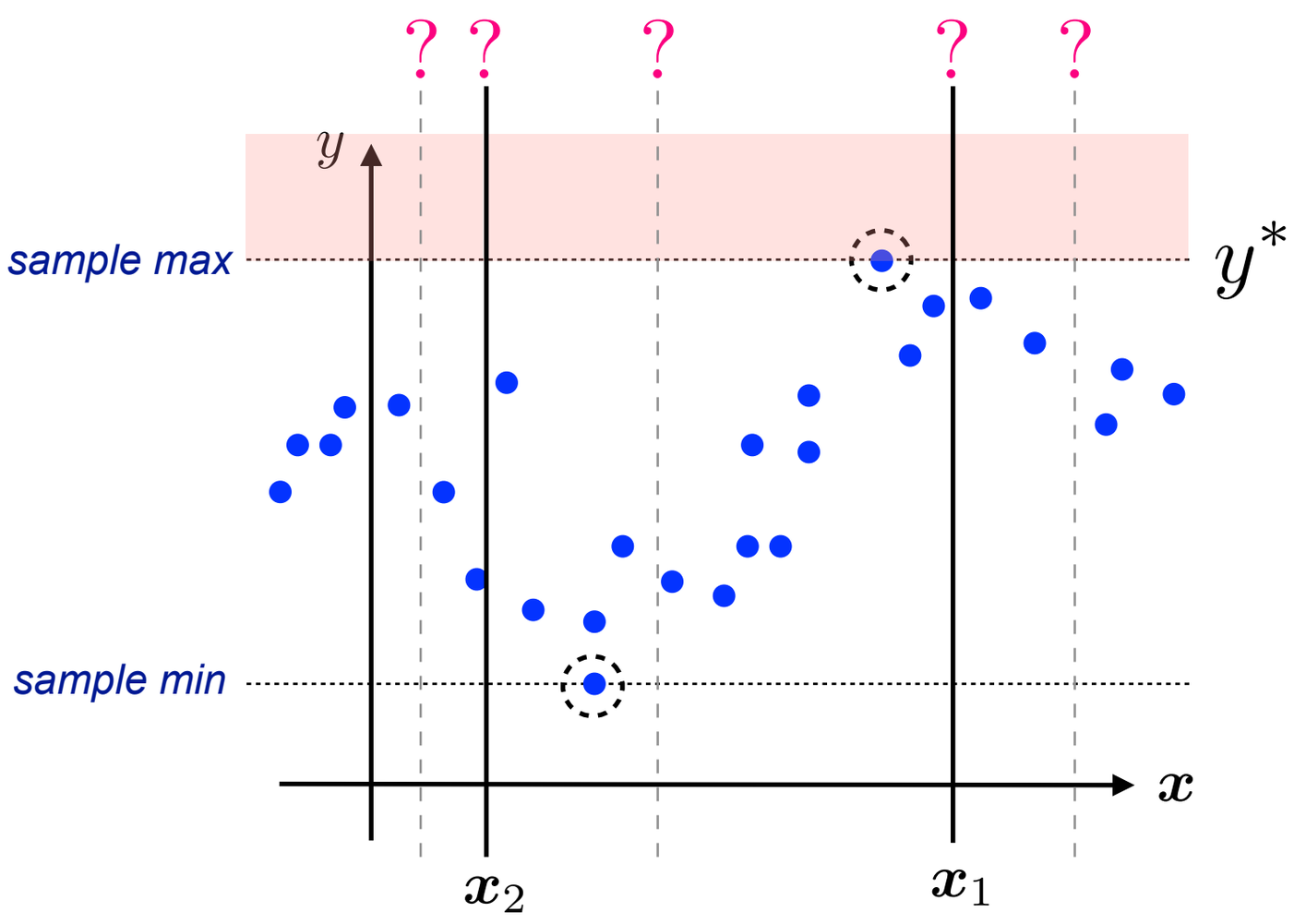
#3. Optimism in the face of uncertainty

We would like to find X better than (hopefully) **the currently known best** y^*.

What next location of $\boldsymbol{x}$ is likely to give higher y?

A fundamental choice: exploitation-exploration tradeoff

Exploitation Make the best decision given current information
Exploration Gather more information by probing uncertain areas



$\boldsymbol{x}_1$ exploitative choice

$\boldsymbol{x}_2$ explorative choice

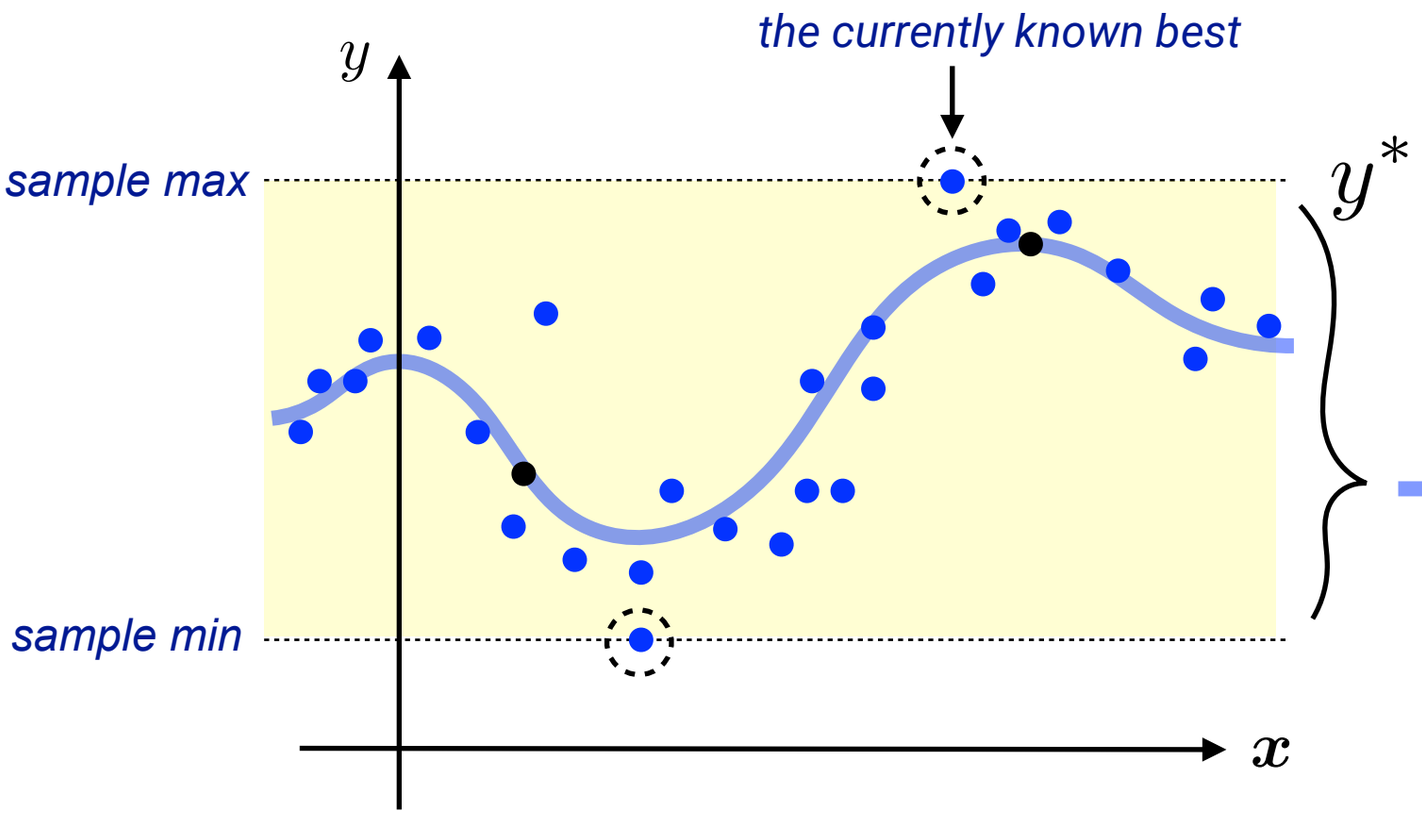
Random choice (e.g. random design) or evenly spaced sampling (e.g. full factorial design) can also work for lower dimensional exploration.

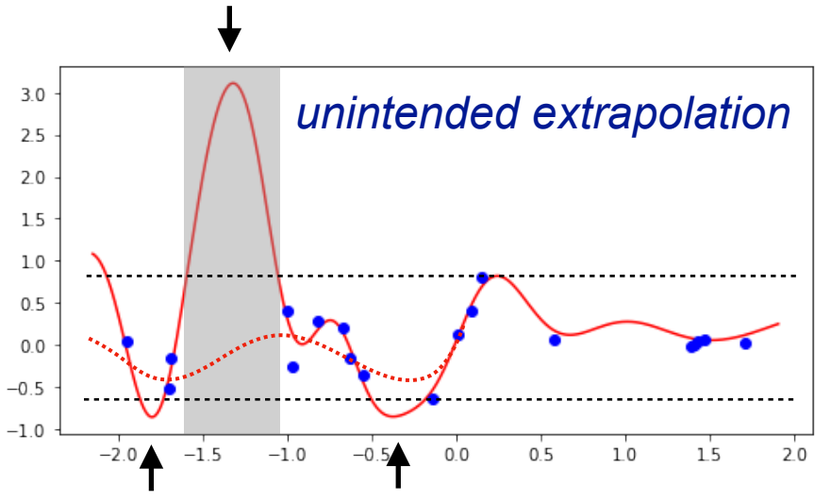
#3. Optimism in the face of uncertainty

We would like to find X better than (hopefully) **the currently known best.**
Now we would like to use ML for **the goal.**

ML fits a function to minimize the average errors, and as a result, ML functions go through the center (mean) of sample output values.

When ML is rightly fitted, the predicted values are **never larger than the known best**, which is inconsistent with the goal.





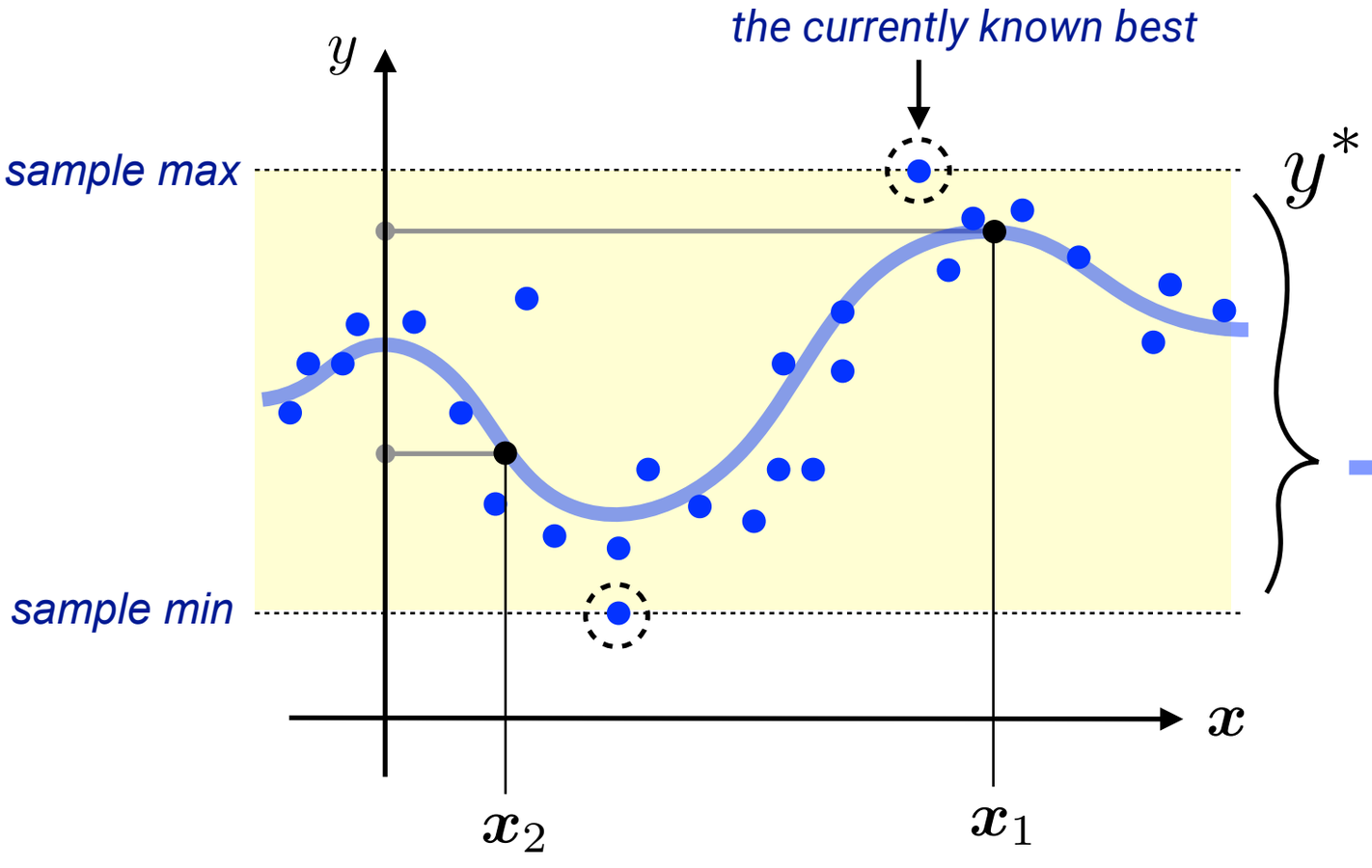
predictions will be groundless when it goes beyond this area

#3. Optimism in the face of uncertainty

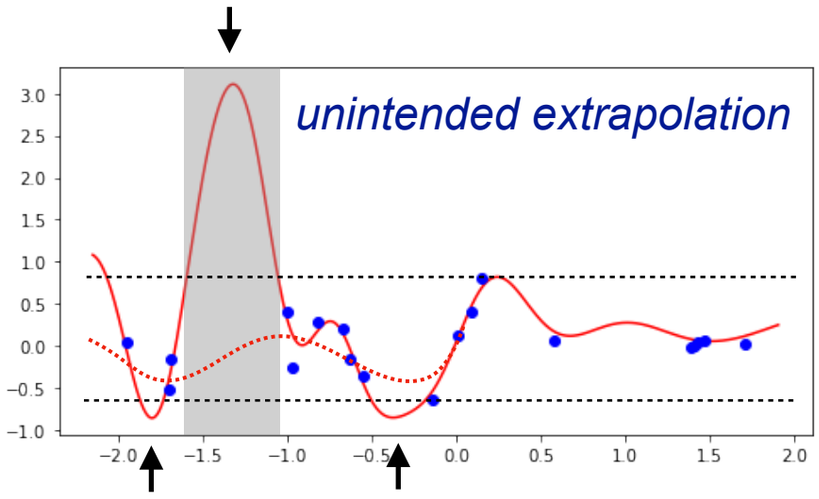
We would like to find X better than (hopefully) **the currently known best.**
Now we would like to use ML for **the goal.**

ML fits a function to minimize the average errors, and as a result, ML functions go through the center (mean) of sample output values.

When ML is rightly fitted, the predicted values are **never larger than the known best**, which is inconsistent with the goal.



goes through here

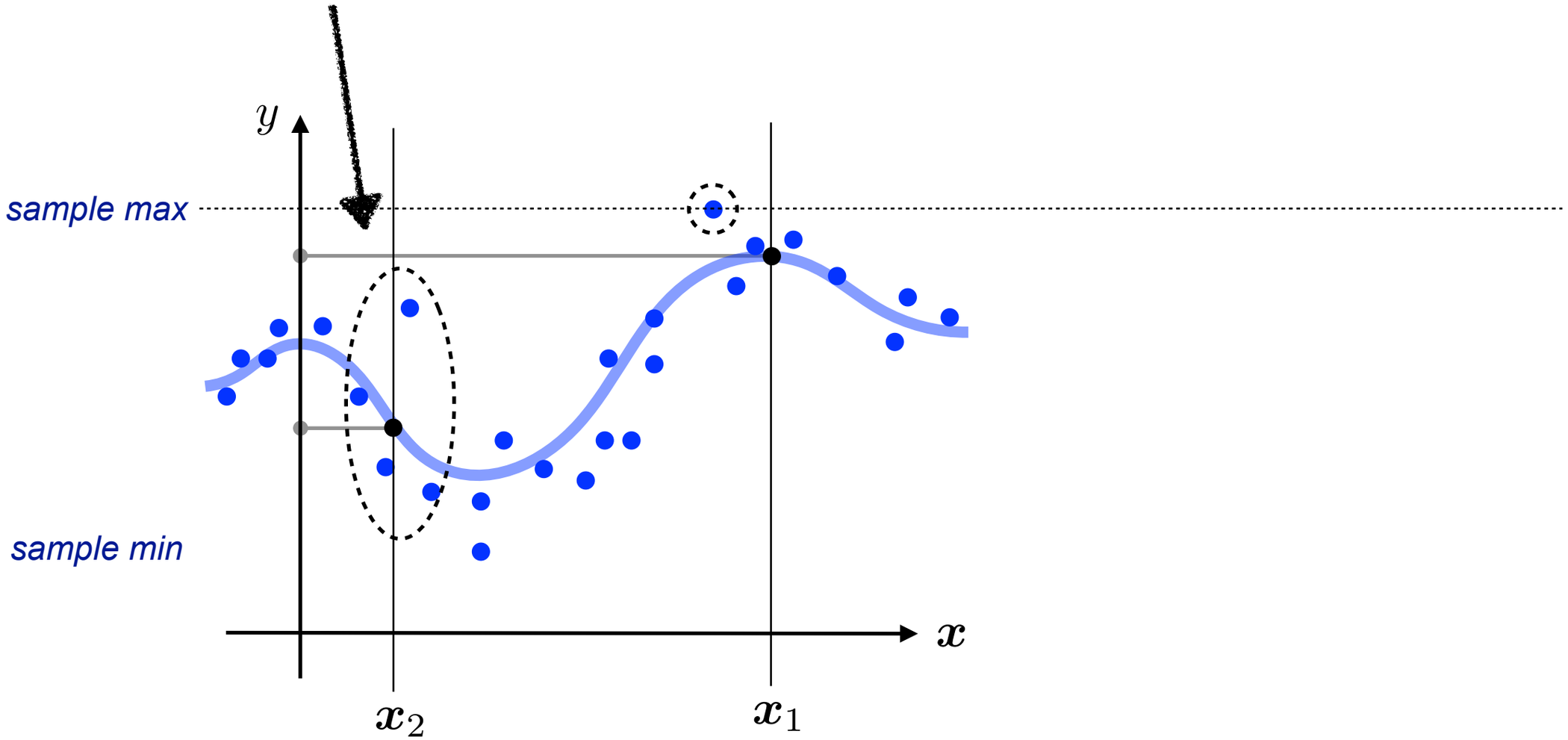


predictions will be groundless when it goes beyond this area

#3. Optimism in the face of uncertainty

This is why we need a criterion taking ***uncertainty*** into consideration instead of direct use of ML predicted values to guide exploration.

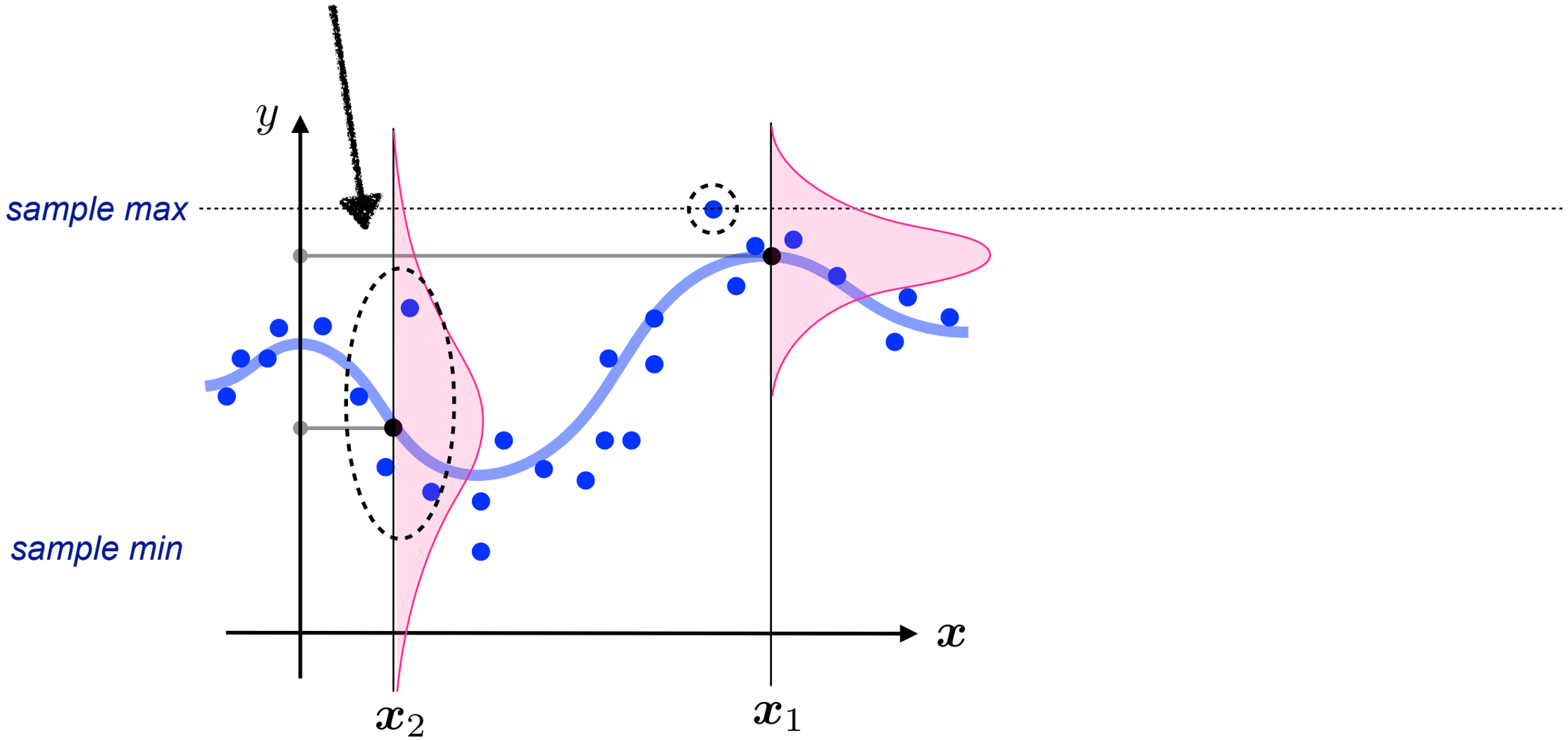
> It'll be nice to gather more information around here even though the mean y is not so high (the predictions have a large variance)



#3. Optimism in the face of uncertainty

This is why we need a criterion taking ***uncertainty*** into consideration instead of direct use of ML predicted values to guide exploration.

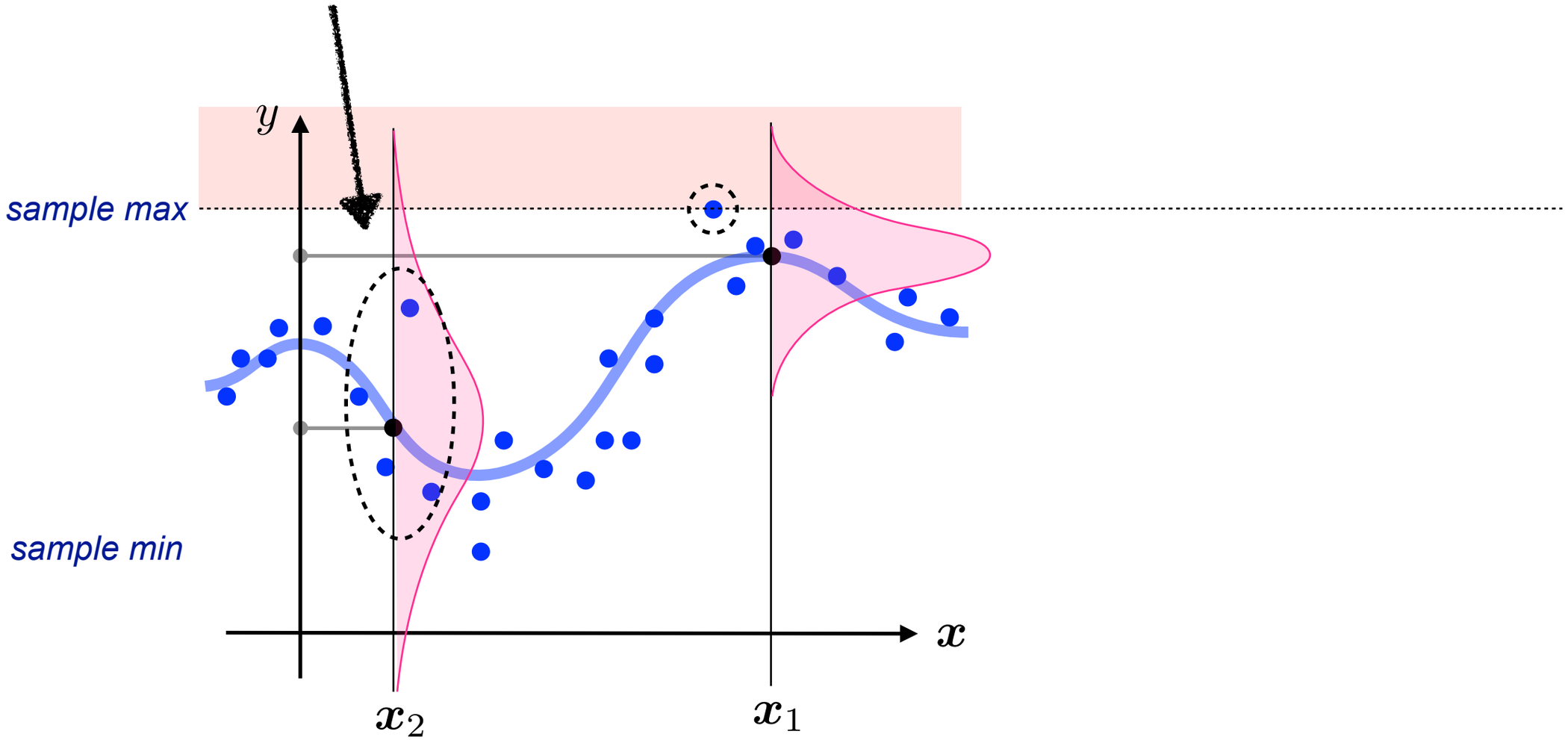
> It'll be nice to gather more information around here even though the mean y is not so high (the predictions have a large variance)



#3. Optimism in the face of uncertainty

This is why we need a criterion taking ***uncertainty*** into consideration instead of direct use of ML predicted values to guide exploration.

> It'll be nice to gather more information around here even though the mean y is not so high (the predictions have a large variance)



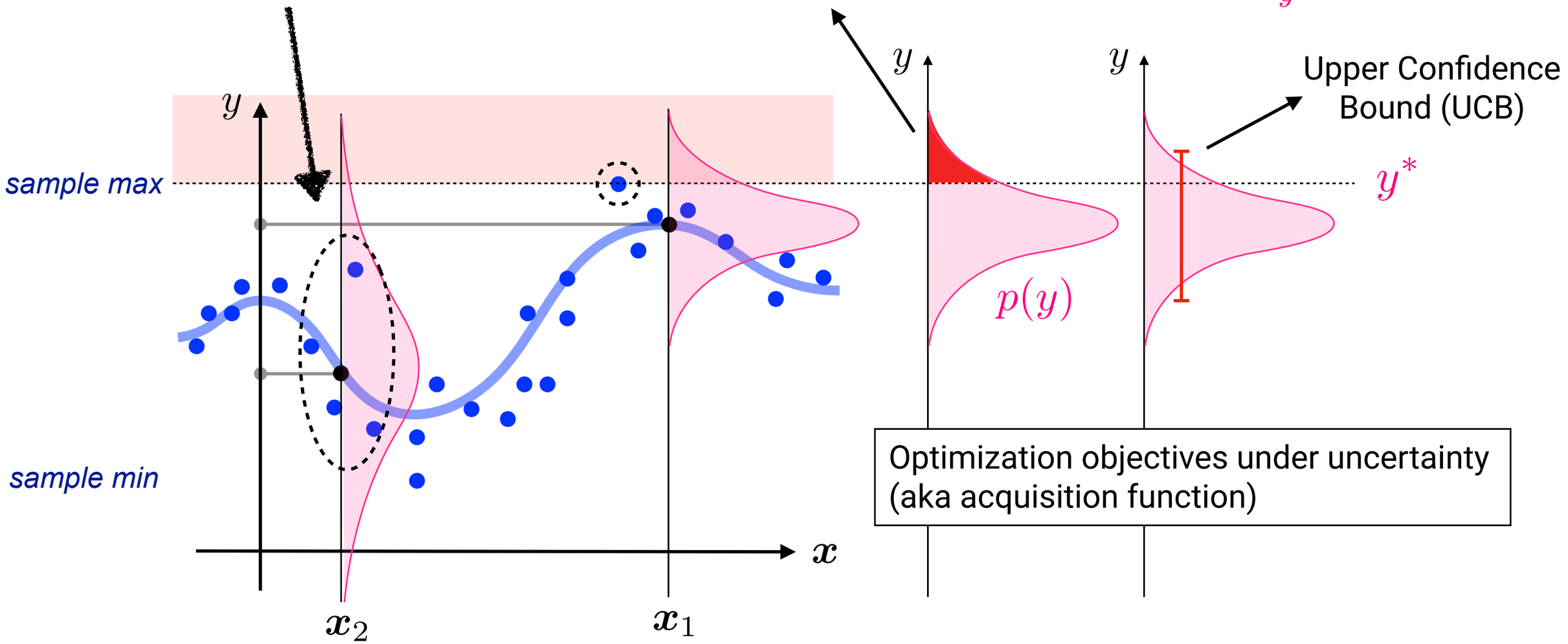
#3. Optimism in the face of uncertainty

This is why we need a criterion taking ***uncertainty*** into consideration instead of direct use of ML predicted values to guide exploration.

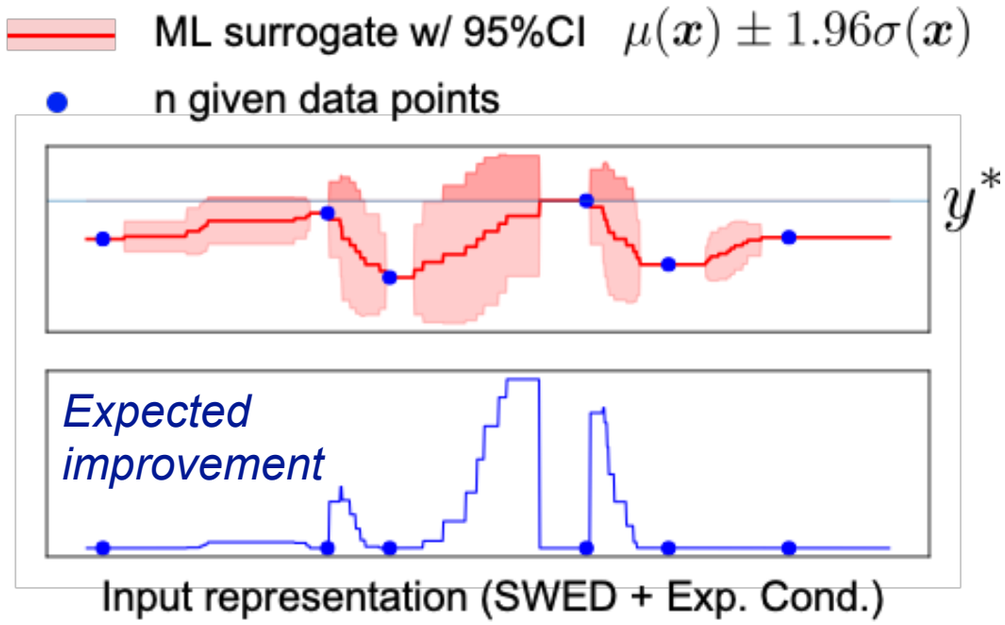
It'll be nice to gather more information around here even though the mean y is not so high (the predictions have a large variance)

Probability of improvement (PI) $P(y > y^*) = \int_{y^*}^{\infty} p(y)dy$

Expected Improvement (EI) $\mathbb{E}[y \mid y > y^*] = \int_{y^*}^{\infty} y \cdot p(y)dy$

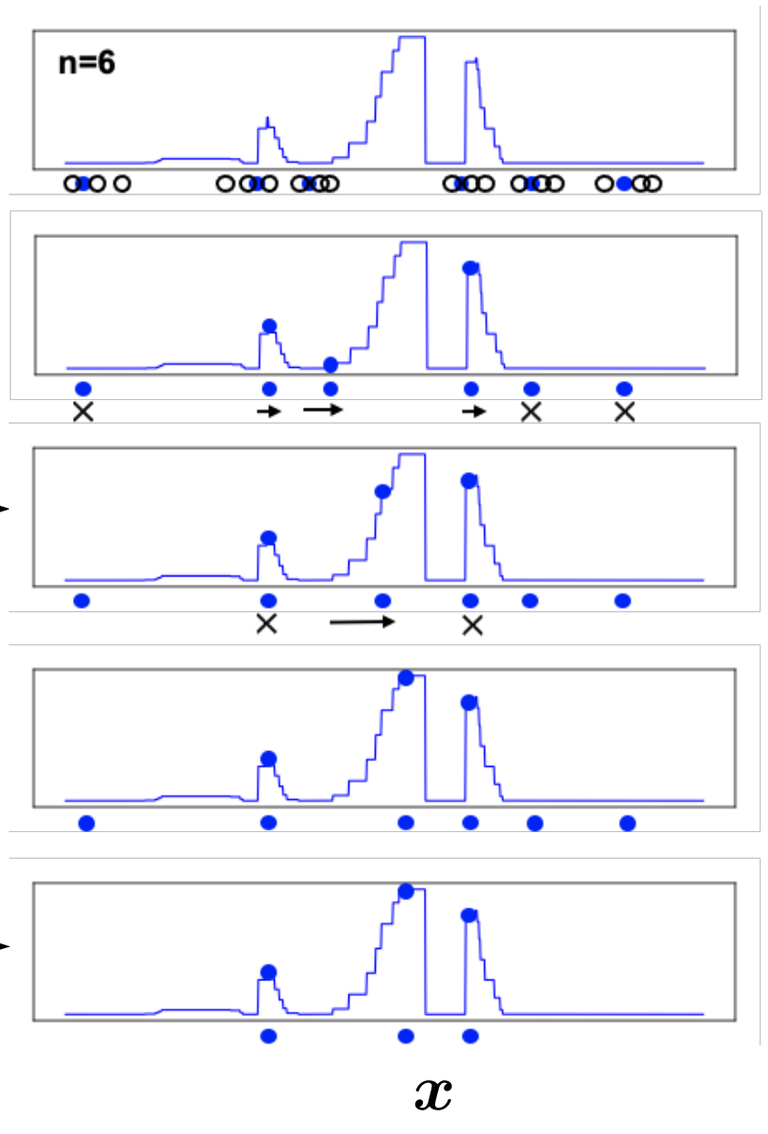


Identifying local peaks of EI of the ML model



Local peaks of EI *would be nice candidates having locally maximal EIs.*

But they are not at given sample points, and the following local search is designed.



multistart from given sample points

adding small random perturbation, and update position when EI increases

stop when local perturbation doesn't change the EI value any more.

run clustering over final candidates, and ***suggest K candidates having locally maximal EI values.***

Every time SWED is changed, the corresponding composition is estimated by our algorithm, and then recalculate valid SWED from it.

Partly because tree ensemble regression functions are locally bumpy, this clustering is effective

Explorative Search with SWED

SWED represent every element with respect to a given set of elemental descriptors.
So we can focus only on the selected elemental properties to explore catalysts.

Each user's intention and focus for catalyst exploration can be design through the **elemental descriptor choice.**

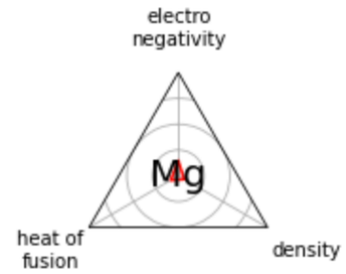
Catalyst: Mg 83.46, Li 16.53

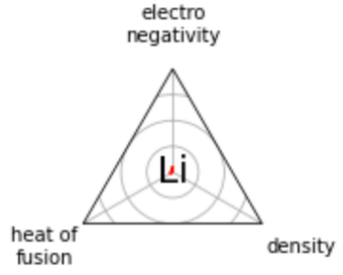
74 elements

Compositional (onehot-like)

Li	Be	B	C	N	F	Na	Mg	Al	Si	...	Ta	W	Re	Os	Ir	Pt	Au	Tl	Pb	Bi
16.53	0.0	0.0	0.0	0.0	0.0	0.0	83.46	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

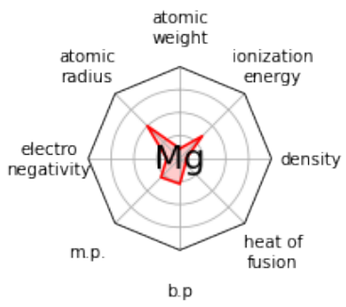
SWED-3

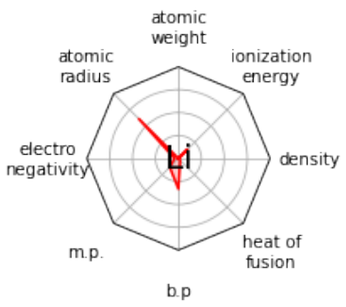
83.46 × 

16.53 × 

0.00 ··· *0.00*

SWED-8

83.46 × 

16.53 × 

0.00 ··· *0.00*

can control specificity & focus

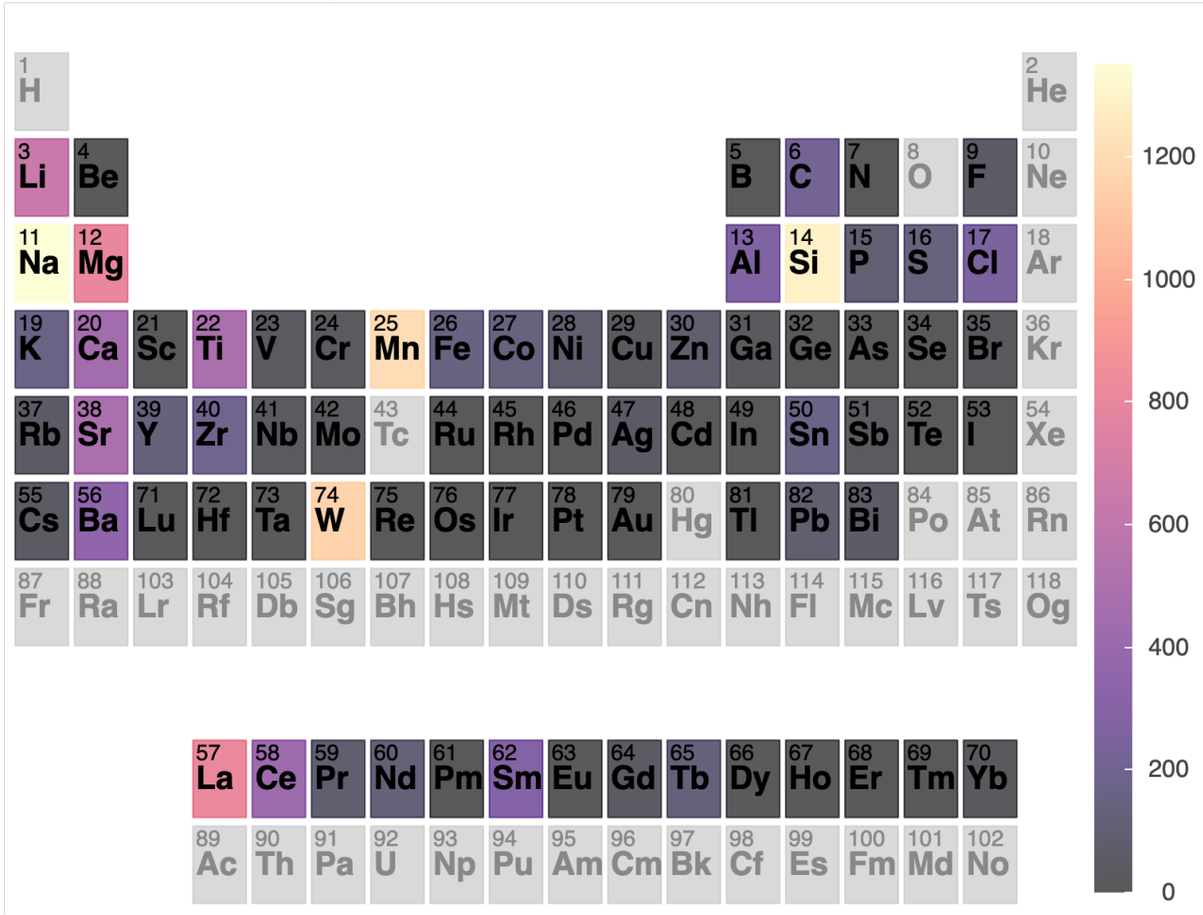
SWED-3 features: electronegativity, density, enthalpy of fusion
SWED-8 features: SWED-3 features + atomic weight, atomic radius, m.p., b.p., ionization enegy

Our updated dataset

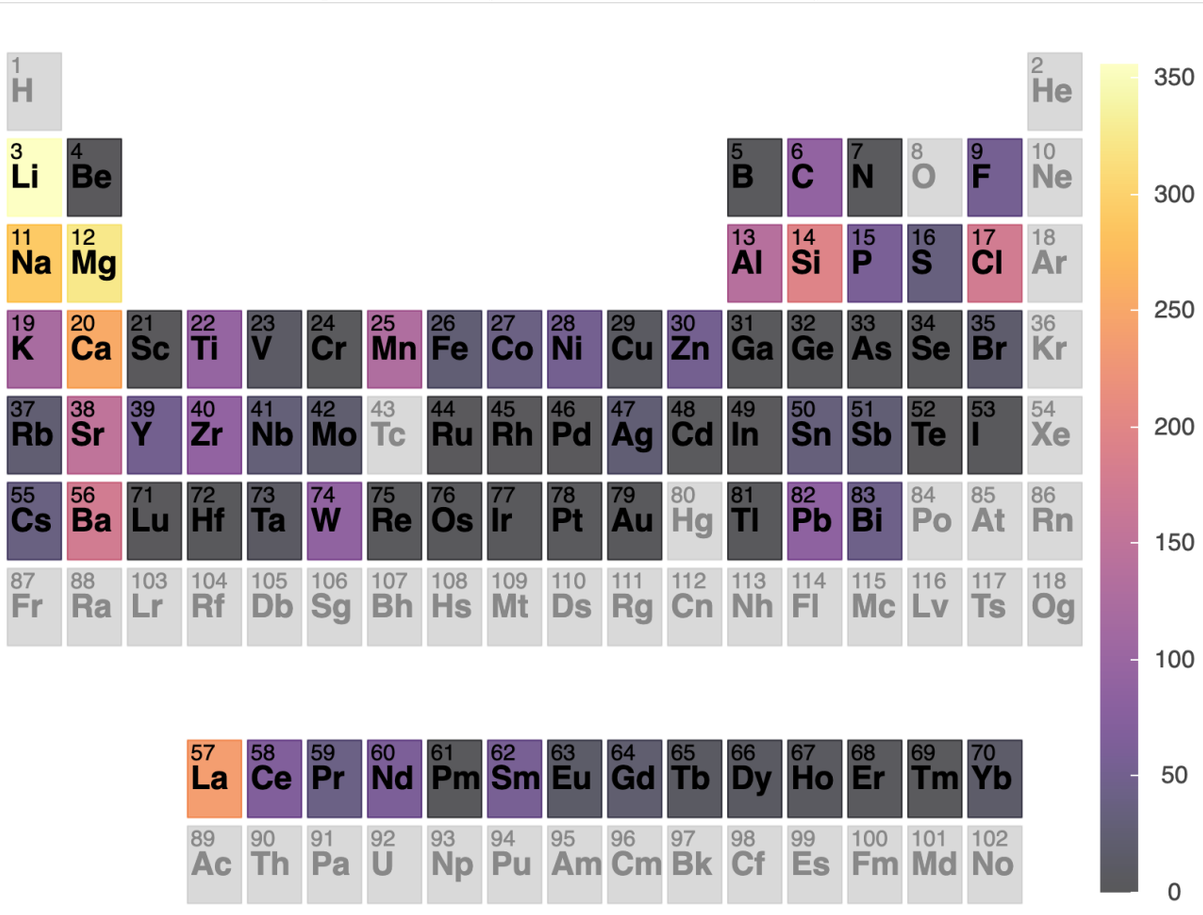
4559 catalyst records from 542 reports

Mine, S.; Takao, M.; Yamaguchi, T.; Toyao, T.*; Maeno, Z.; Hakim Siddiki, S. M. A.; Takakusagi, S.; Shimizu, K.*; Takigawa, I.* *ChemCatChem* 2021. https://doi.org/10.1002/cctc.202100495.

The update dataset:
4559 catalyst records
from 542 reports (2010 - 2019)



The original dataset:
1866 catalyst records
from 421 reports (1982 - 2009)



ML Predictions of C_2 yields

1. **Conventional:** composition + condition
2. **Proposed(Exploitative):** composition + SWED + condition
3. **Proposed(Explorative):** SWED + condition w/ SWED→composition estimator

Table 2. Comparison of prediction accuracy (RMSE) and coefficient of determination (R^2) for the C_2 yields (%) of the OCM reaction. Two datasets and three ML methods were tested using 10-fold cross-validation. The numbers shown in parentheses are the corresponding σs.

	Pre-2010 dataset			Entire OCM dataset		
ML model	RFR	ETR	XGB	RFR	ETR	XGB
Conventional Method						
Training Error [%]	1.66 (0.02)	0.17 (0.03)	1.07 (0.37)	1.50 (0.02)	0.75 (0.03)	2.21 (0.38)
Test Error [%]	4.50 (0.38)	4.65 (0.50)	4.34 (0.34)	3.66 (0.23)	3.65 (0.20)	3.71 (0.23)
Test R^2	0.536	0.504	0.567	0.713	0.716	0.706
Proposed Method (Exploitative)						
Training Error [%]	1.63 (0.02)	0.17 (0.02)	0.55 (0.31)	1.50 (0.02)	0.76 (0.03)	1.73 (0.26)
Test Error [%]	4.39 (0.43)	4.30 (0.52)	4.25(0.41)	3.66 (0.27)	3.52 (0.25)	3.58 (0.28)
Test R^2	0.557	0.575	0.583	0.713	0.736	0.722
Proposed Method (Explorative) with all the 8 descriptors						
Training Error [%]	1.68 (0.02)	0.17 (0.02)	0.29 (0.10)	1.52 (0.02)	0.76 (0.03)	1.32 (0.31)
Test Error [%]	4.50 (0.48)	4.44 (0.50)	4.43 (0.52)	3.70 (0.29)	3.57 (0.27)	3.56 (0.28)
Test R^2	0.536	0.547	0.547	0.708	0.727	0.728
Proposed Method (Explorative) with 3 descriptors[a]						
Training Error [%]	1.66 (0.02)	0.17 (0.02)	0.34 (0.14)	1.52 (0.02)	0.76 (0.03)	1.27 (0.18)
Test Error [%]	4.45 (0.34)	4.45 (0.35)	4.41 (0.35)	3.69 (0.30)	3.63 (0.26)	3.56 (0.27)
Test R^2	0.547	0.540	0.556	0.709	0.717	0.728

[a] The electronegativity, density, and $\triangle H_{fus}$ were used as descriptors.

RFR (Random Forest); ETR (ExtraTrees); XGB (XGBoost)
SWED-3 features: electronegativity, density, enthalpy of fusion
SWED-8 features: SWED-3 features + atomic weight, atomic radius, m.p., b.p., ionization enegy

Top 20 highest-EI candidates based on SWED-3

Table 3. 20 most promising candidate catalyst systems for further testing in the OCM, as suggested using SMBO with ETR coupled with the proposed method (explorative) and the entire dataset. In addition to EIs, the predicted values μ, standard deviations σ, and 95 % confidence intervals (as $\mu \pm 1.96\sigma$) are also shown. Oxygen is not shown in the elemental compositions. Only three descriptors, electronegativity, density, and $\triangle H_{fus}$, were used.

Elemental composition	Promoter	Preparation method	T [K]	$P(CH_4)/P(O_2)$	P_{total} [bar]	Contact time [s]	EI	Predicted C_2 yield [%] mean	sd	95 %CI lower	95 %CI upper
Mn:72.3 Li:27.7	B	Solid-phase technique	1023	1.67	1.01	1.20	2.29	25.69	13.52	−0.80	52.19
Sr:50.0 Ce:45.0 Yb:5.0	–	Solid-phase technique	1023	1.99	1.01	0.79	2.29	25.88	13.35	−0.28	52.04
Si:60.9 Na:19.3 Cl:17.2 Mn:1.6 W:1.0	–	Hydrothermal treatment	1023	1.60	1.01	0.02	1.67	25.55	11.71	2.60	48.50
Mn:80.0 Li:20.0	Cl	Solid-phase technique	1023	1.96	1.00	0.60	1.37	23.63	12.12	−0.13	47.40
Mg:82.8 Li:17.2	–	Solid-phase technique	1043	3.00	1.01	5.50	1.31	22.97	12.38	−1.29	47.23
C:42.7 Sc:23.6 Ge:17.8 K:10.1 I:5.8	–	Physical mixing	1023	1.60	1.01	0.00	1.29	24.00	11.56	1.34	46.66
Si:45.9 Mg:22.0 Ru:20.5 Sc:6.0 Ge:5.7	–	Physical mixing	1023	1.62	1.39	0.03	1.22	23.48	11.69	0.57	46.39
Ge:30.9 Sc:30.7 As:25.3 Be:6.7 I:6.3	–	Physical mixing	1023	1.41	2.75	0.13	1.05	22.75	11.56	0.11	45.40
Si:35.5 Br:32.4 Mg:14.4 Al:9.0 Ho:6.5 Y:2.1	S	Physical mixing	1023	1.28	1.01	0.00	1.01	25.54	9.49	6.94	44.13
Mg:36.0 Ge:33.6 Mo:30.3	–	Precipitation	1073	2.50	1.01	3.60	0.95	23.97	10.34	3.71	44.23
La:46.4 Ge:27.9 Cu:25.7	–	Ceramic method	1023	1.94	0.68	2.40	0.95	23.16	10.86	1.88	44.44
Sc:36.9 Ca:32.9 Mo:30.2	–	Ceramic method	1040	0.90	0.70	1.79	0.92	23.84	10.31	3.63	44.05
Nd:83.6 Ge:16.4	–	Hydrothermal treatment	1100	3.95	2.24	0.35	0.92	22.20	11.38	−0.10	44.50
Si:34.4 Ca:29.1 Ge:23.2 Nb:9.7 As:3.6	–	Physical mixing	1002	1.39	1.90	0.02	0.91	20.33	12.55	−4.26	44.92
C:45.8 Sc:20.8 Ge:20.3 Mo:7.7 Nb:5.3	–	Physical mixing	1015	0.89	2.96	0.04	0.91	20.79	12.22	−3.16	44.75
Sc:49.4 Au:34.1 Ge:16.5	–	Ceramic method	1023	2.00	1.01	2.00	0.90	22.62	11.00	1.06	44.19
C:38.8 Sc:31.5 Ge:16.0 As:7.9 Rh:5.8	–	Physical mixing	1017	0.63	3.02	0.07	0.89	21.32	11.83	−1.88	44.51
Mo:38.9 V:37.9 Ge:23.2	–	Ceramic method	1048	2.00	0.85	2.03	0.89	22.39	11.14	0.57	44.22
Sr:45.7 Ge:33.7 As:20.6	–	Ceramic method	1023	1.99	0.69	1.20	0.89	22.39	11.14	0.56	44.22
Sc:38.7 Mo:37.5 Ca:23.9	–	Ceramic method	1023	1.96	1.01	4.10	0.88	21.29	11.80	−1.85	44.42

As appeared not included in the data

Fs, Se, Os, Bm infrequent elements also observed

↓

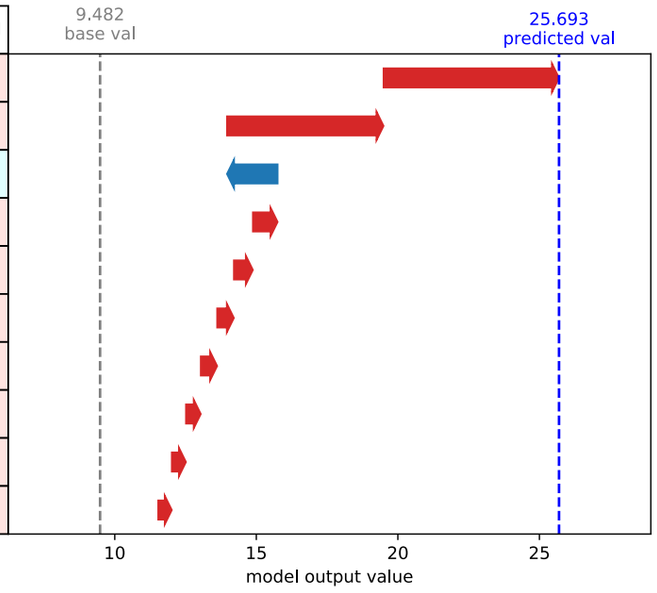
though these are toxic and impractical but explorative suggestions were able be made

Post analysis for models and suggested catalysts

With SHAP, feature importance/permutation importance, dependency plot, etc.

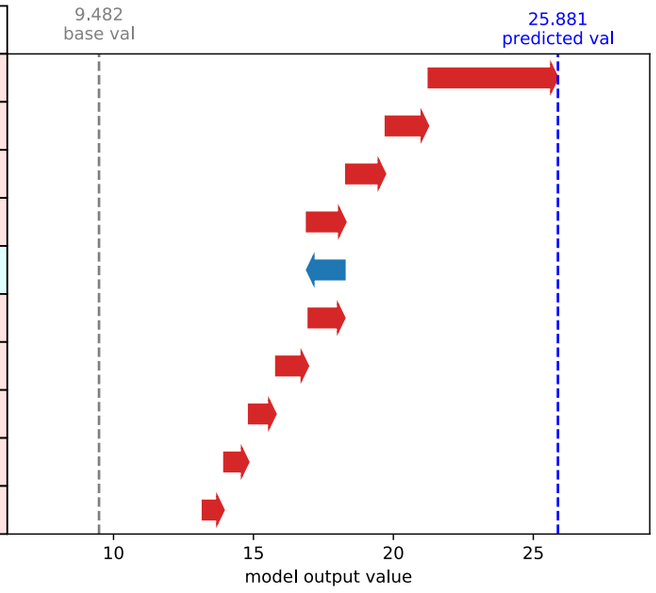
1st: (1) Mn: 72.3 (2) Li: 27.7

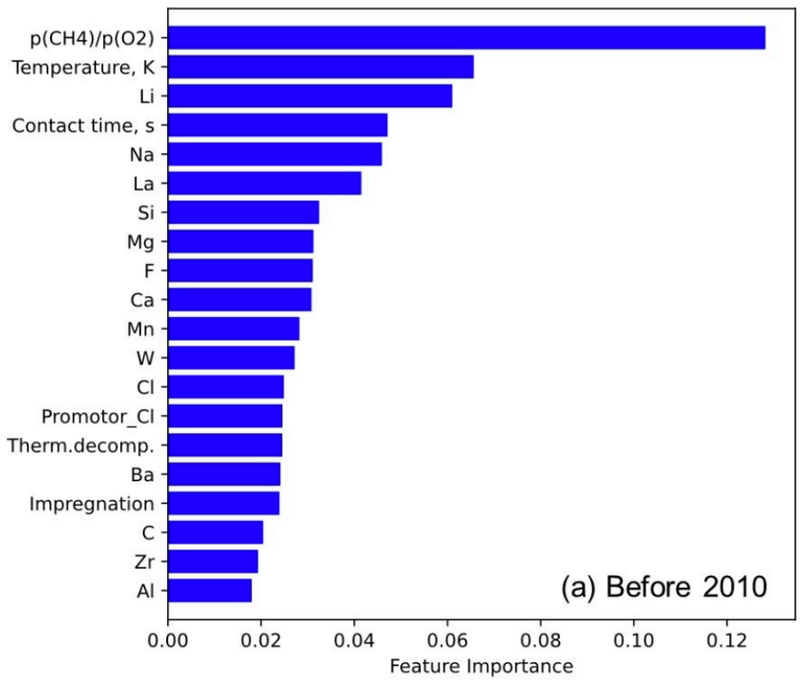
	feat name	feat val	SHAP val
1	p(CH4)/p(O2)	1.667	6.198
2	Promotor_B	1.0	5.531
3	Solid-phase technique	1.0	-1.79
4	density (2)	14.792	0.874
5	delta fus H (1)	933.393	0.671
6	Therm.decomp.	0.0	0.591
7	Temperature, K	1023.0	0.579
8	electronegativity (2)	26.869	0.523
9	p total, bar	1.013	0.5
10	Impregnation	0.0	0.48

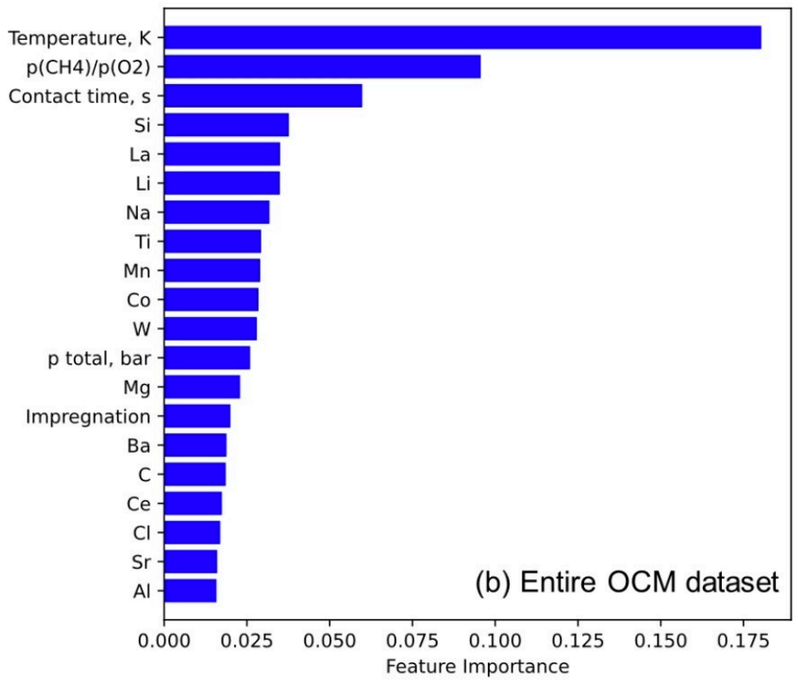


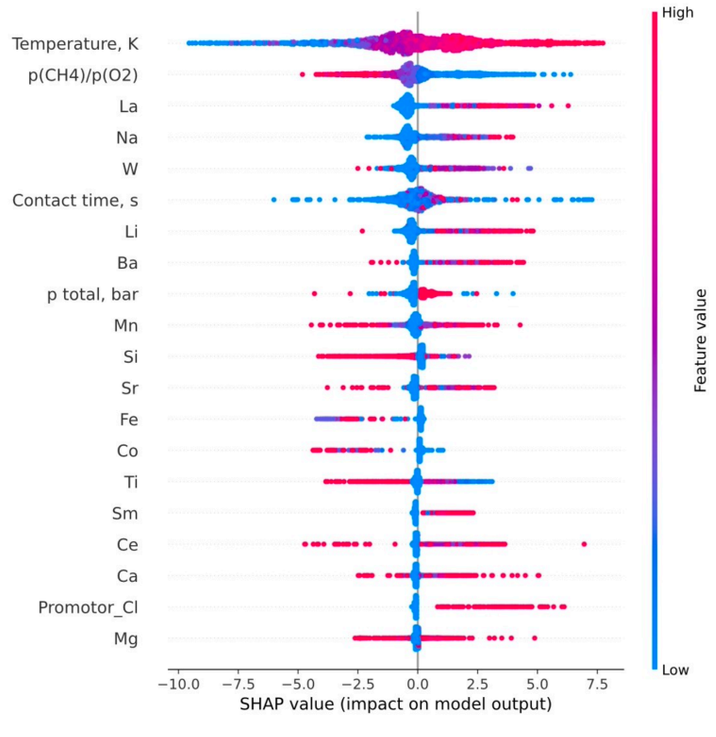
2nd: (1) Sr:50.0 (2) Ce:45.0 (3) Yb:5.0

	feat name	feat val	SHAP val
1	p(CH4)/p(O2)	1.985	4.628
2	electronegativity (1)	49.5	1.535
3	density (3)	34.5	1.415
4	delta fus H (1)	371.5	1.401
5	Solid-phase technique	1.0	-1.363
6	electronegativity (3)	5.3	1.301
7	density (2)	304.65	1.16
8	Contact time, s	0.79	0.973
9	density (1)	132.0	0.882
10	p total, bar	1.013	0.768









Today's talk

Our struggles for better ML practices with underspecified, sparse, biased observational data (i.e. a collection of experimental facts from literature)

ChemCatChem
The European Society Journal for Catalysis
Chemistry Europe
European Chemical Societies Publishing

Analysis of Updated Literature Data up to 2019 on the Oxidative Coupling of Methane Using an Extrapolative Machine-Learning Method to Identify Novel Catalysts

Dr. Shinya Mine, Motoshi Takao, Taichi Yamaguchi, Dr. Takashi Toyao ✉, Dr. Zen Maeno, Dr. S. M. A. Hakim Siddiki, Dr. Satoru Takakusagi, Prof. Ken-ichi Shimizu ✉, Dr. Ichigaku Takigawa ✉

First published: 31 May 2021 | https://doi.org/10.1002/cctc.202100495

ChemCatChem
The European Society Journal for Catalysis
Chemistry Europe
European Chemical Societies Publishing

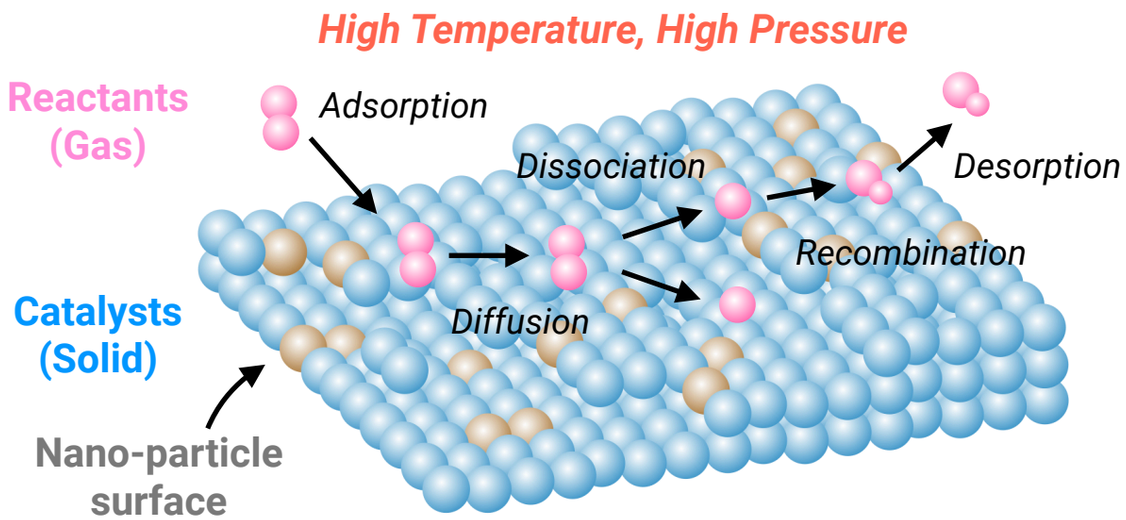
Statistical Analysis and Discovery of Heterogeneous Catalysts Based on Machine Learning from Diverse Published Data

Keisuke Suzuki, Dr. Takashi Toyao, Dr. Zen Maeno, Dr. Satoru Takakusagi, Prof. Ken-ichi Shimizu ✉, Dr. Ichigaku Takigawa ✉

First published: 09 July 2019 | https://doi.org/10.1002/cctc.201900971 | Citations: 10

Gas-phase reactions on sold-phase catalyst surface (Heterogeneous catalysis)

Industrial Synthesis (e.g. Haber-Bosch), Automobile Exhaust Gas Purification, Methane Conversion, etc.



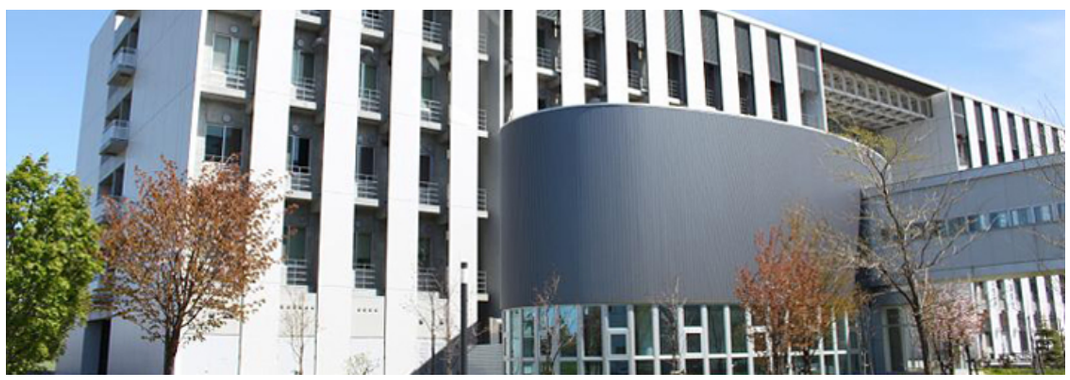
Devilishly complex too-many-factor process!!

God made the bulk;
the **surface** was invented by the **devil**

—— Wolfgang Pauli

Acknowledgements

Institute for Catalysis ICAT
北海道大学 触媒科学研究所
Hokkaido University, Institute for Catalysis



Ken-ichi SHIMIZU

Satoru TAKAKUSAGI

Takashi TOYAO

Zen MAENO

科研費 KAKENHI

CREST

統合物質創製化学研究推進機構 IRCCS Integrated Research Consortium on Chemical Sciences

wpi World Premier International Research Center Initiative

Keisuke SUZUKI
Motoshi TAKAO
Shinya MINE
Taichi YAMAGUCHI
S. M. A. Hakim Siddiki