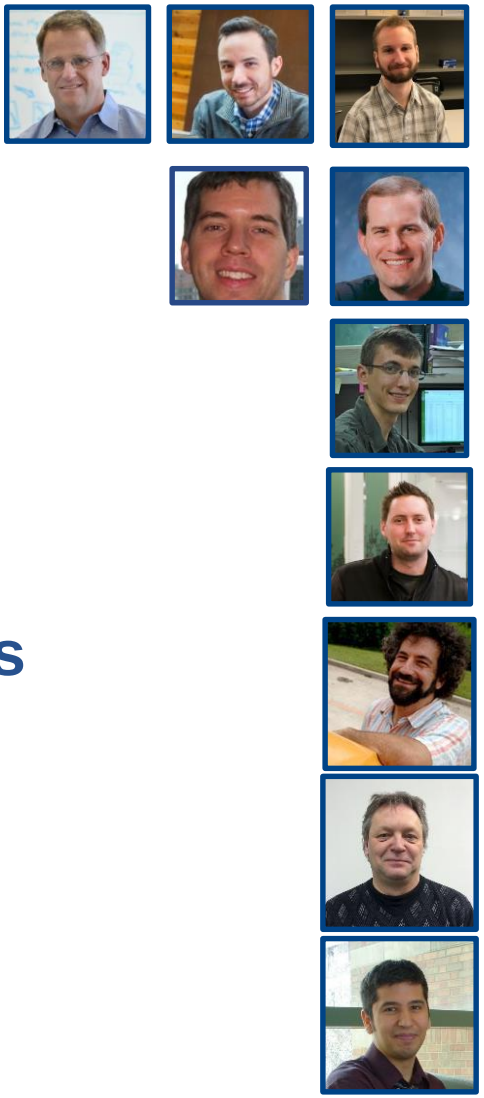
Class Overview

- **Materials Data Facility**
  - Data publication
  - Data discovery
  - Data service integration
  - Automation
- **Overview of ML/DL in Materials Science**
- **Model Serving with DLHub**
- **Lots of time for questions, so please ask!**

MDF Overview



Build data services to

- Empower researchers to publish data, regardless of size, type, and location
- Automate data and metadata ingest, to enable capture of many valuable materials datasets
- Enable unified search and discovery across disparate materials data sources

Deploy with APIs to simplify connection to other data efforts and to enable automation

CHiMaD NIST

MDF Connect - Connecting Community Services

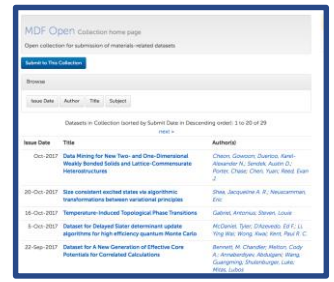
- Make it easy to deposit into many services from one location
- Strictly opt-in for cross-posting datasets

Submit Data [UI or API]

Enrich Data

Send to Community

- Mint DOIs
- Associate metadata
- Persist datasets



- Query
- Browse
- Aggregate

Forge

build passing pypi v0.4.6

CREATE

Inputs

Google Drive HTEM

4CeeD MaterialsCommons

NIST MML DSpace

box CDCS

MIDWEST BIG DATA HUB

MATERIALS DATA FACILITY

MDF Connect

MDF Connect

Extract / Transform

MATERIALS DATA FACILITY

Publish / Discover

NIST MRR

CITRINE INFORMATICS

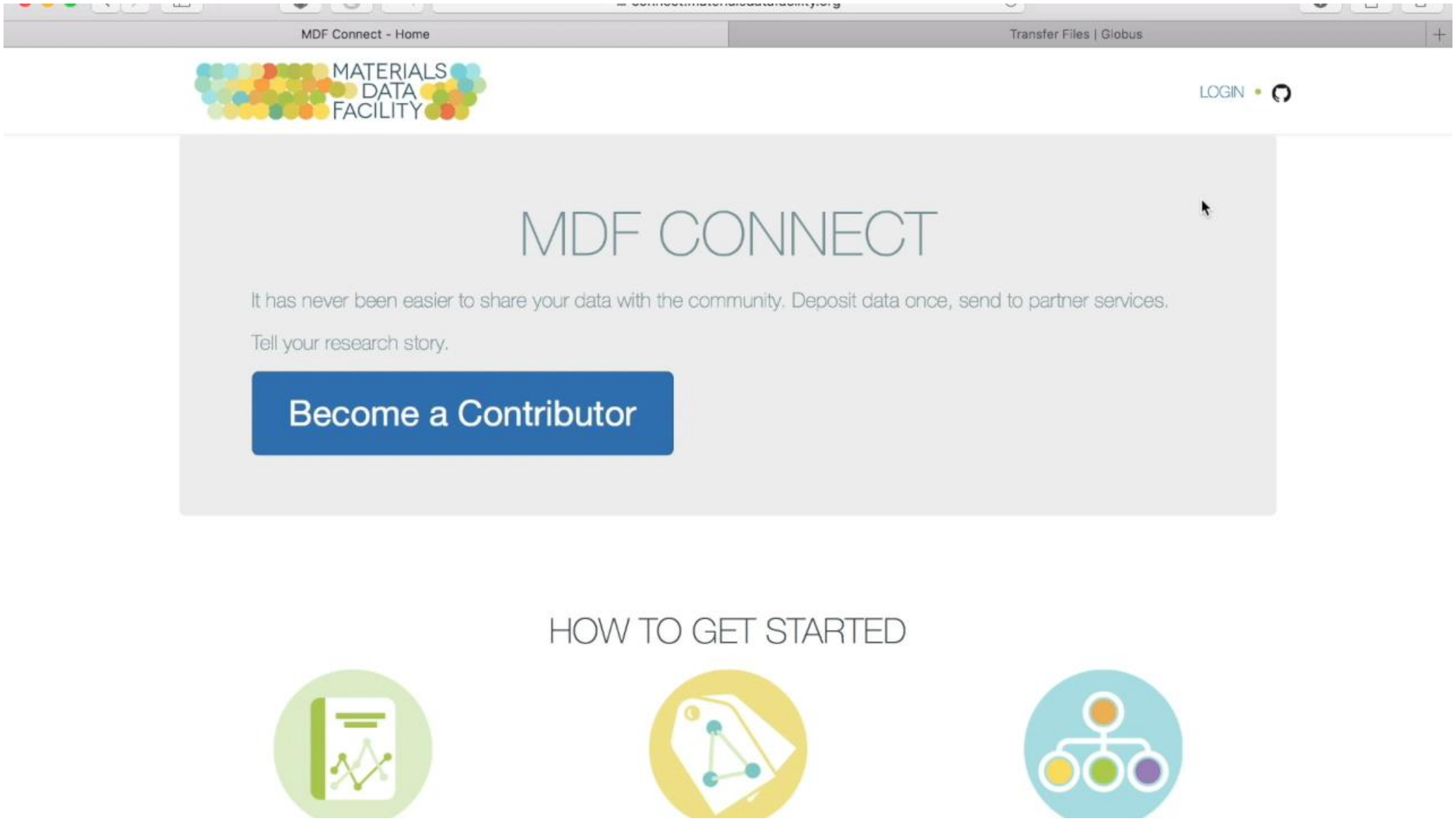
In progress

NanoMine

...

CHiMaD NIST

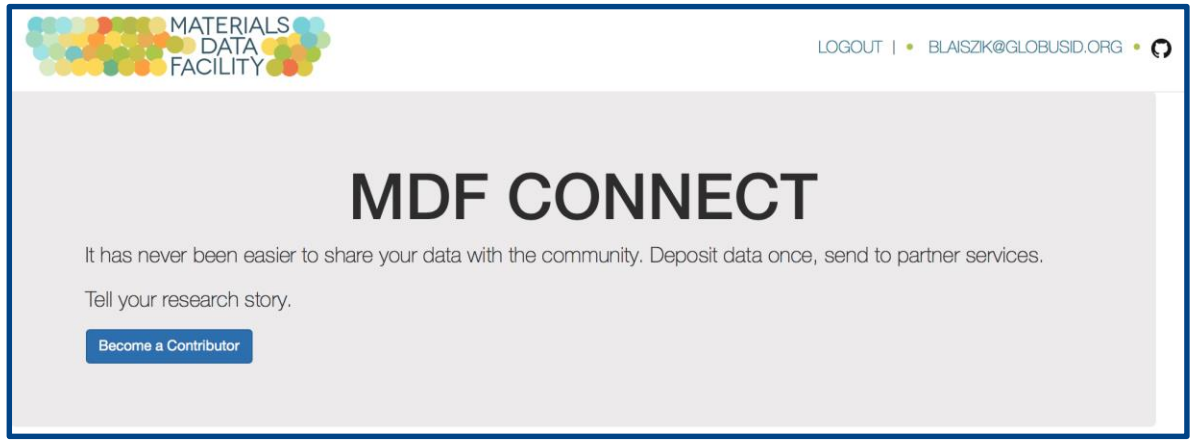
MDF Connect Prototype

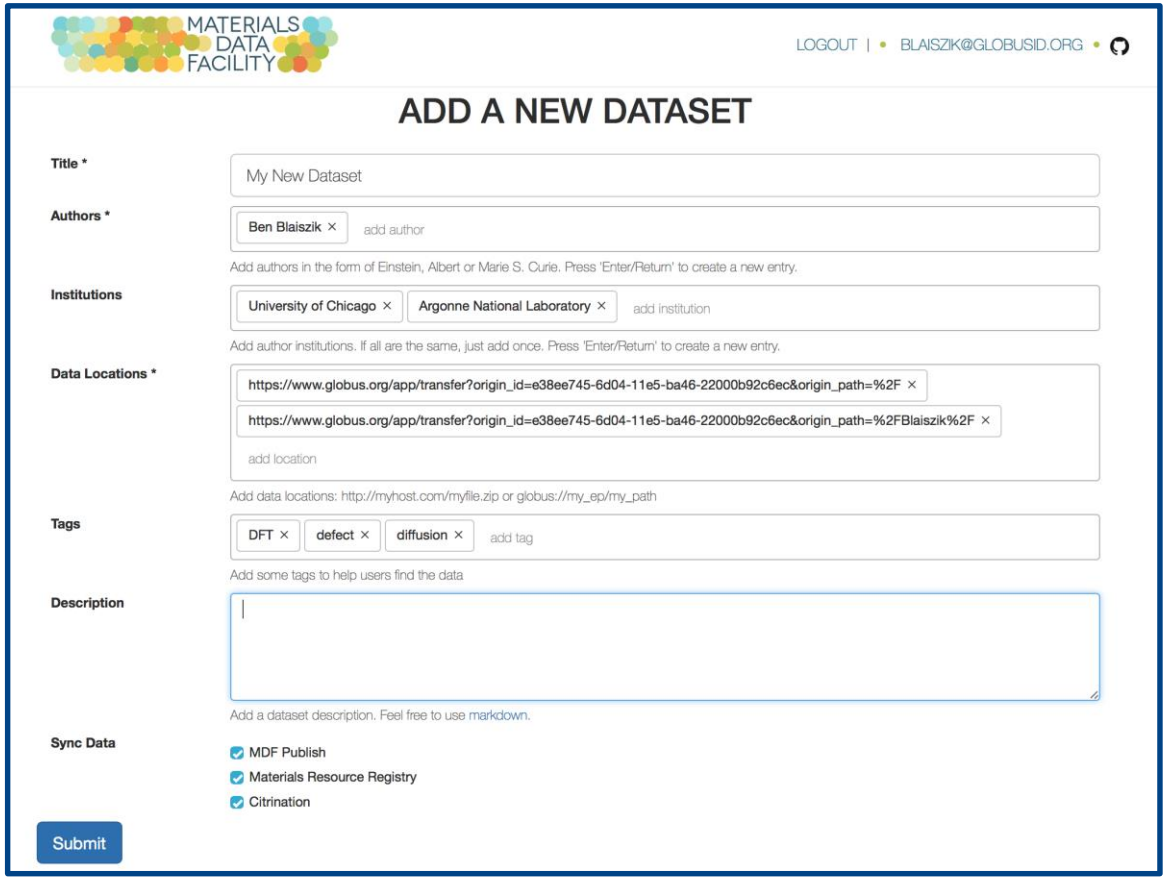


MDF Connect Alpha

MDF Connect Alpha invites going out next week

- Functionality includes dataset indexing (MDF), dataset publication (MDF), MRR registration, Citrine submission
- Send data from Globus, Google Drive, or HTTP using Python or Web UI





CHiMaD NIST

MDF Connect Alpha

MDF Connect Alpha invites going out next week

- Functionality includes dataset indexing (MDF), dataset publication (MDF), MRR registration, Citrine submission
- Send data from Globus, Google Drive, or HTTP using Python or Web UI

```
pip install mdf_connect_client
```

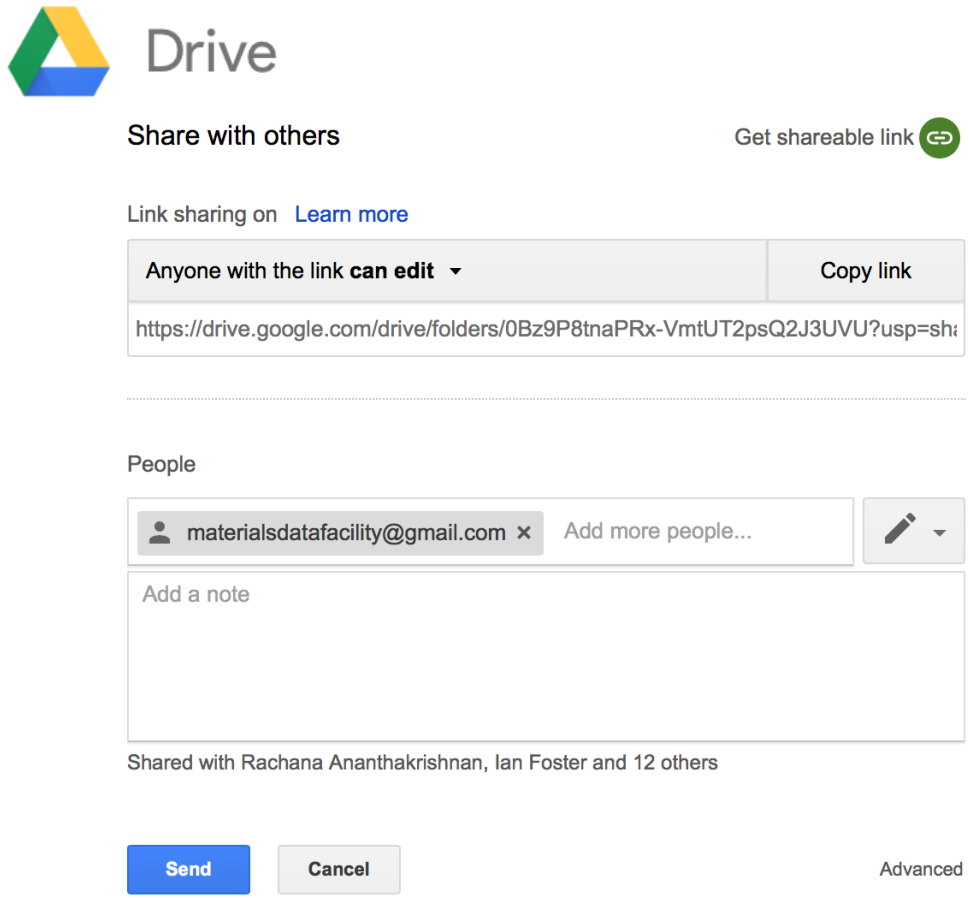
```
from mdf_connect_client import MDFConnectClient

mdf = MDFConnectClient()

authors = ["Blaiszik, Ben", "Jonathon Gaff"]
affs = ["University of Chicago", "Argonne National Laboratory"]

mdf = MDFConnectClient(title="My Dataset Title",
                       authors=authors,
                       affiliations=affs)

mdf.add_data(["googledrive:///mydata.zip", "globus://1a2b3c/data/"])
mdf.add_services("mdf_publish", "mrr", "citrine")
mdf.submit()
```

```
mdf.create_mrr_block({"dataOrigin": "experiment"})
mdf.mrr
```

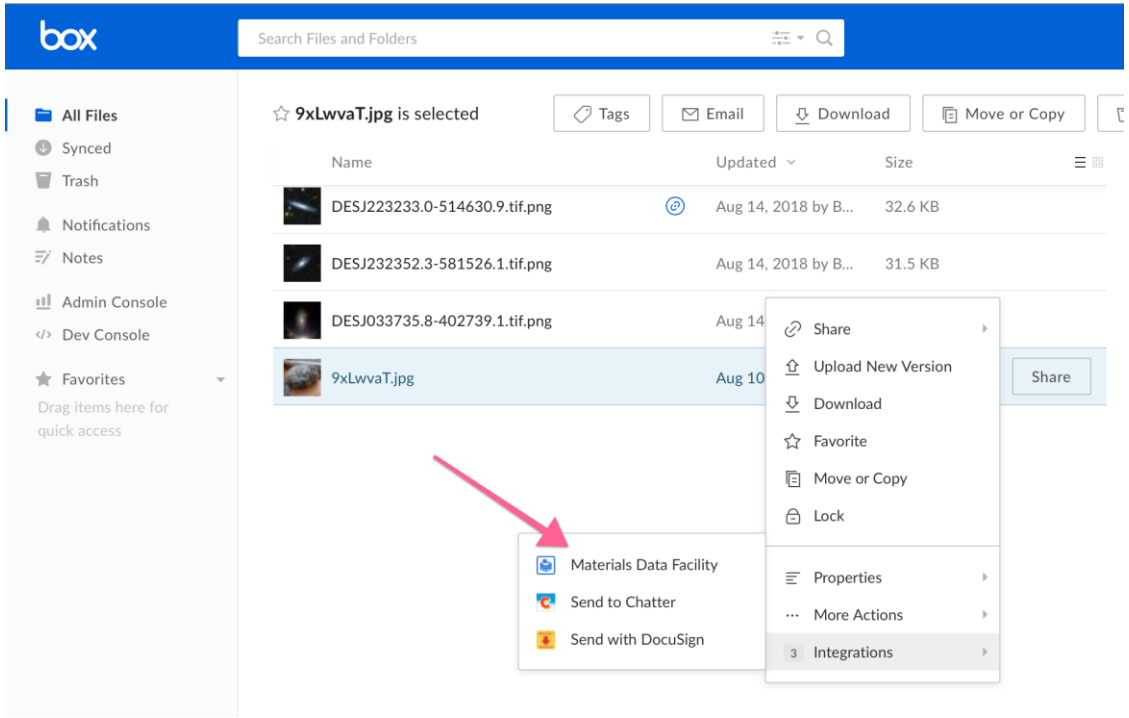
CHiMaD NIST

MDF Connect Input Integration Highlights

Google Drive (operational)



Box (prototype)



DropBox (operational)

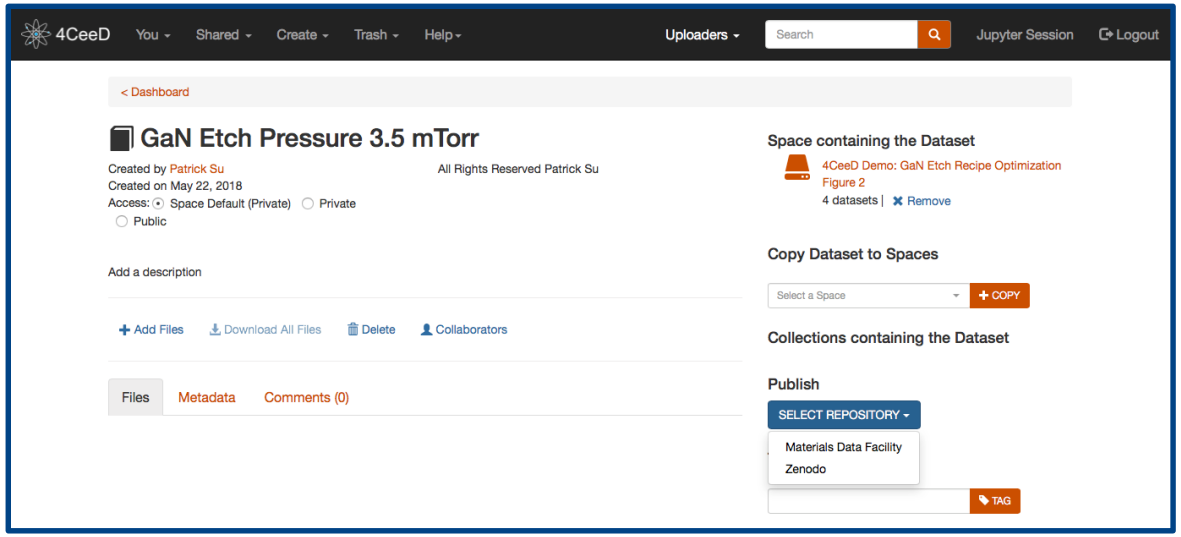
Figshare (operational)

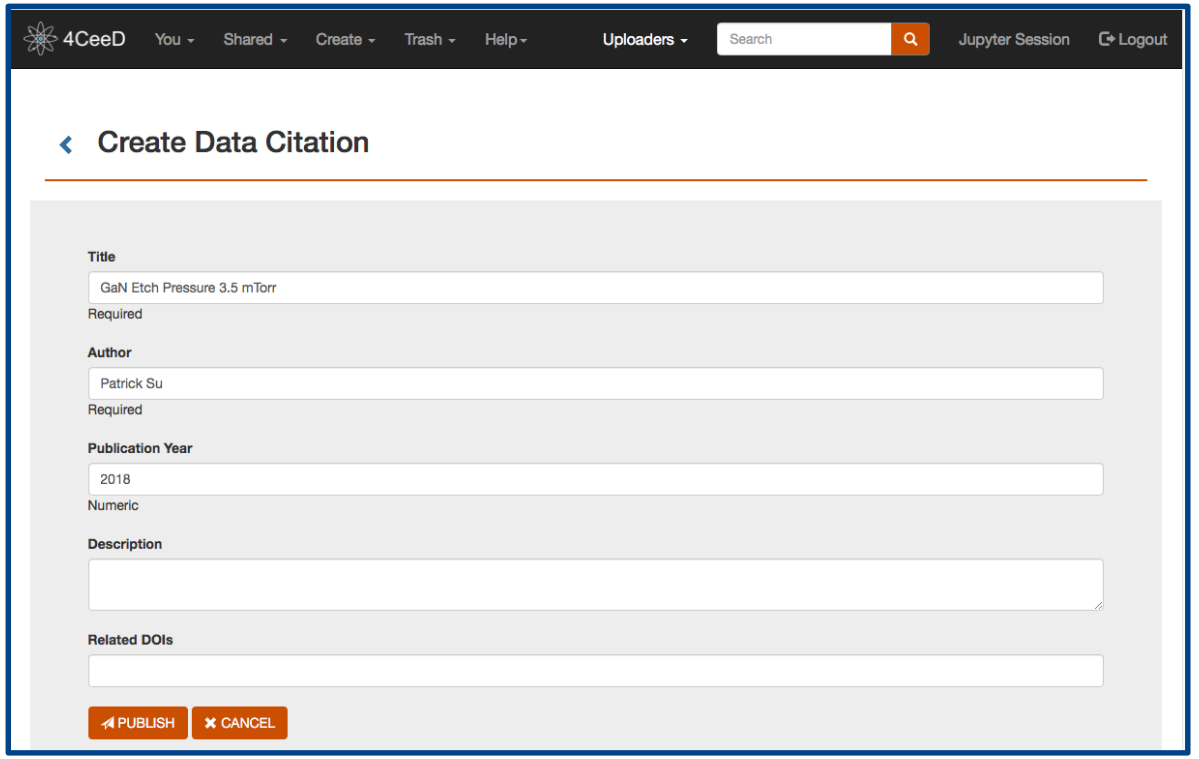
w/ Ben Galewksky @ UIUC

CHiMaD NIST MIDWEST BIG DATA HUB

MDF Connect Input Integration Highlights

4CeeD





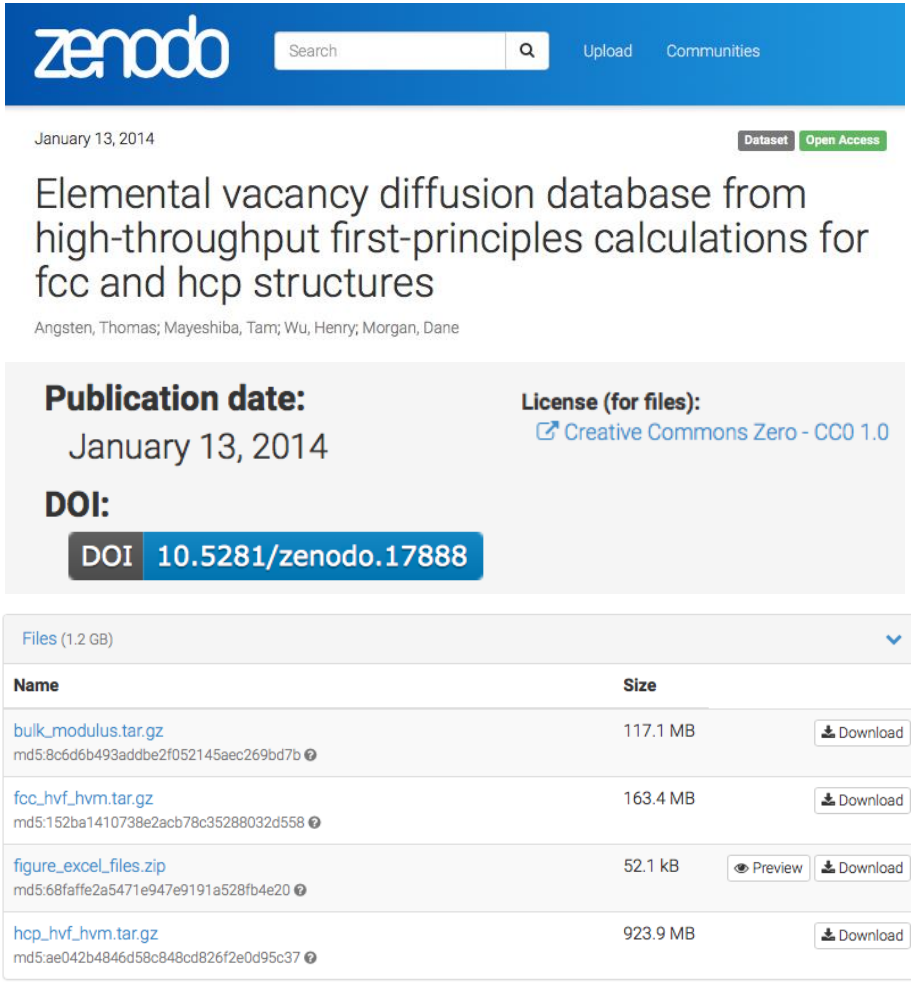
CHiMaD NIST MIDWEST BIG DATA HUB

w/ Ben Galewksky @ UIUC

Materials Data Discovery

Start your search here

- Data are often locked:
  - In archives (zip, tar, etc.) [e.g., Zenodo, NIST DSpace]
  - Behind web forms [many research group resources]
  - Within clunky, partially functional APIs
- Data are located in many distributed locations
- Data file formats are heterogeneous
  - How would you generally search by material composition??



MDF Forge

https://github.com/materials-data-facility

Forge

pypi v0.5.1 | build passing | coverage 85%

Forge is the Materials Data Facility Python package to interface and leverage the MDF Data Discovery service. Forge allows users to perform simple queries and facilitiates moving and synthesizing results.

Installation

```
pip install mdf_forge
```

Compile records for a larger, mixed source, result set

```
elements = ["Al"]
sources = ["khazana_vasp", "sluschi", "ab_initio_solute_database"]
my_ep = "c8ee7e5c-6d04-11e5-ba46-22000b92c6ec"
my_path = "/Users/ben/Desktop/blaiszik-macbookpro/dft_training_set"

mdf = Forge()
res = mdf.search_by_elements(elements=elements, sources=sources, limit=9999)
mdf.get_globus(res, dest=my_path,
               local_ep=my_ep, preserve_dir=True)
```

```
Processing records: 100%|██████████| 10/10 [00:00<00:00, 19.05it/s]
Submitting transfers: 100%|██████████| 1/1 [00:00<00:00,  3.60it/s]
All transfers submitted
Submission IDs: 3fbfc637-7181-11e7-a9fd-22000bf2d287
```

Materials Informatics Skunkworks (D. Morgan)

matminer

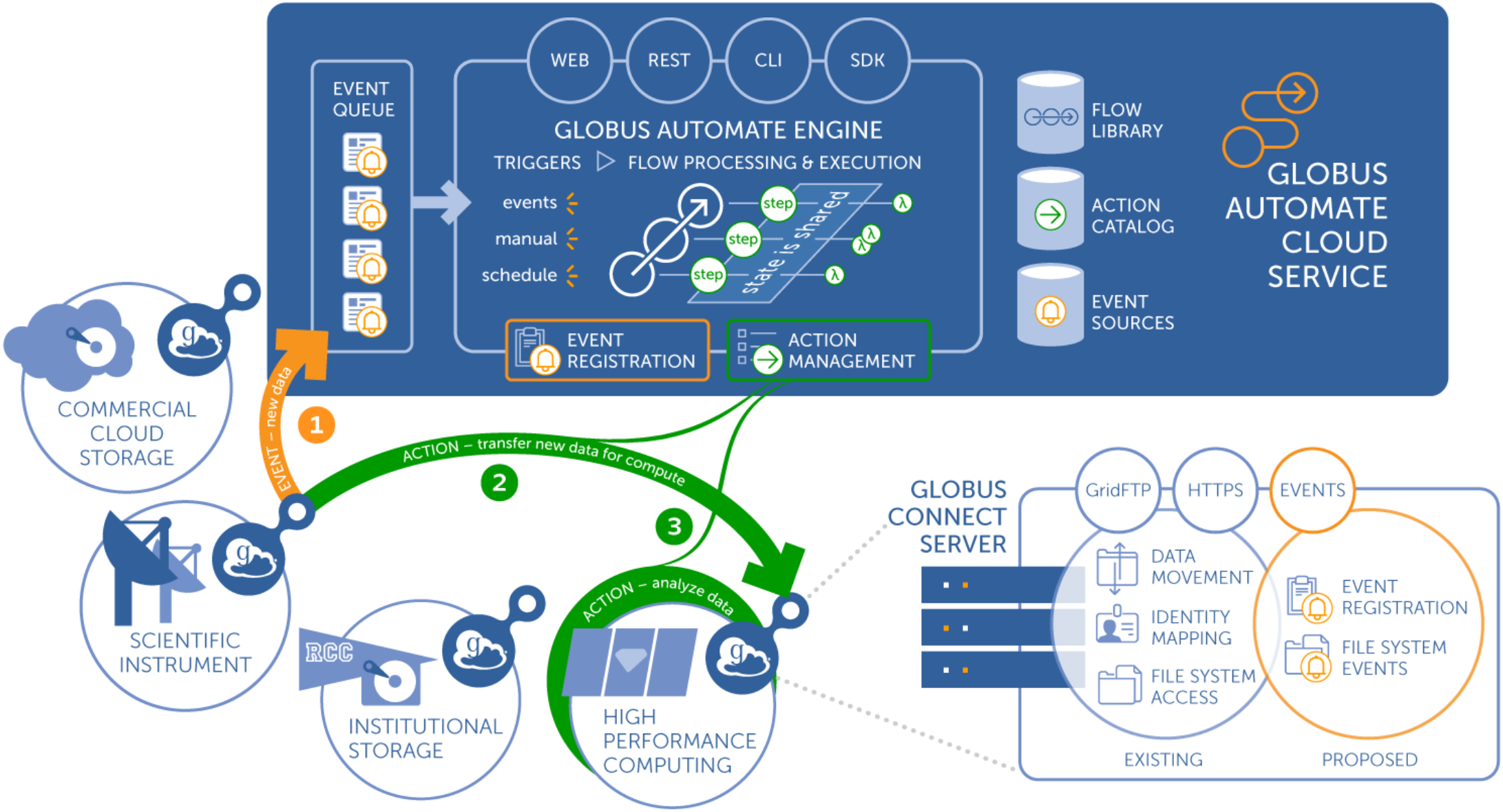
- Data mining and featurizing in materials science
- Integrated to pull data from MDF

(L. Ward, A. Jain)

NATIONAL DATA SERVICE PARTICIPANT

NDS Labs Workbench

Automation in Science



Some Automation Actions

Auth

- User login
- Secure service interactions

Identify

- Manage namespace
- Mint DOI

Search

- Catalog
- Query

MATERIALS DATA FACILITY

Transfer

- Transfer data
- Set permission

Execute

parsl

- Remote execution
- Self optimization

Describe

MATERIALS DATA FACILITY

- Gather metadata
- Validate metadata

Curate

MATERIALS DATA FACILITY

- Solicit input
- Edit / Approve

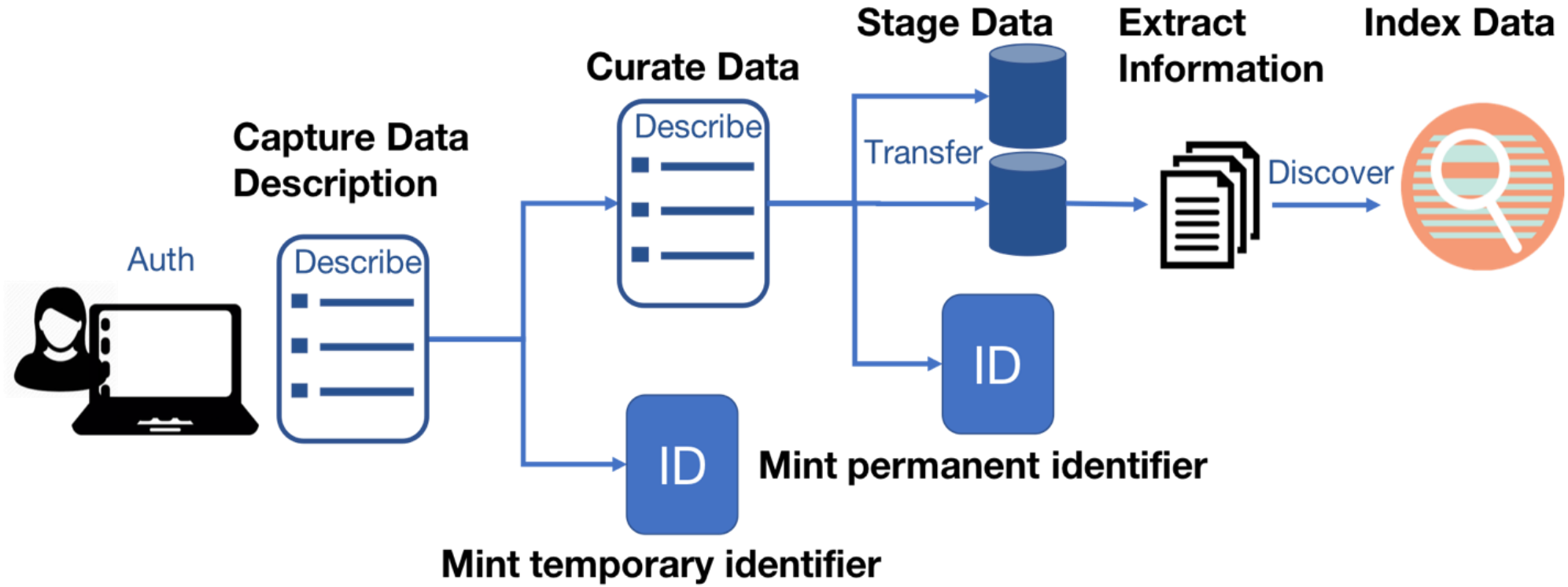
Extract

MATERIALS DATA FACILITY

- Automate metadata extraction
- Invoke container extraction

CHiMaD NIST

Data Publication with Automate



CHiMaD NIST

Deep Learning and Materials Design

Logan Ward
Postdoctoral Scholar
Data Science and Learning Division, ANL
Department of Computer Science, University of Chicago
18 October 2018

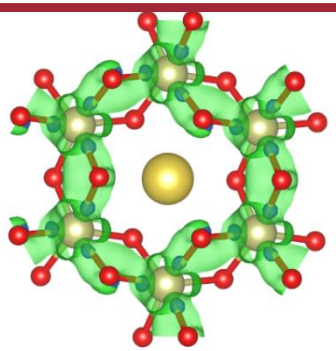
globus labs

I'm a Materials Scientist

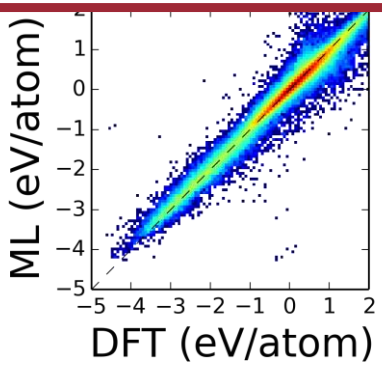
Affiliation: Post-doc with Ian Foster since Jan. 2017

Background: PhD in Materials Science, Northwestern
BS/MS in Materials Science, from The Ohio State University

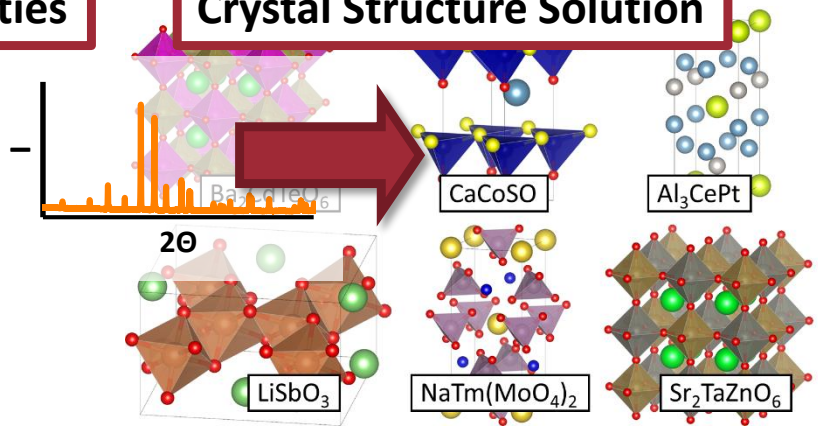
Density Functional Theory



ML of Material Properties



Crystal Structure Solution



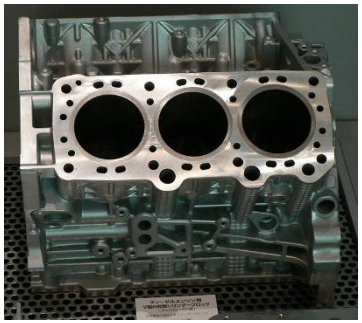
Why UC/ANL?

1. Wanted to see ML in other fields
2. Build tools for others to use

MATERIALS DATA FACILITY

Materials Engineering: A narrow introduction

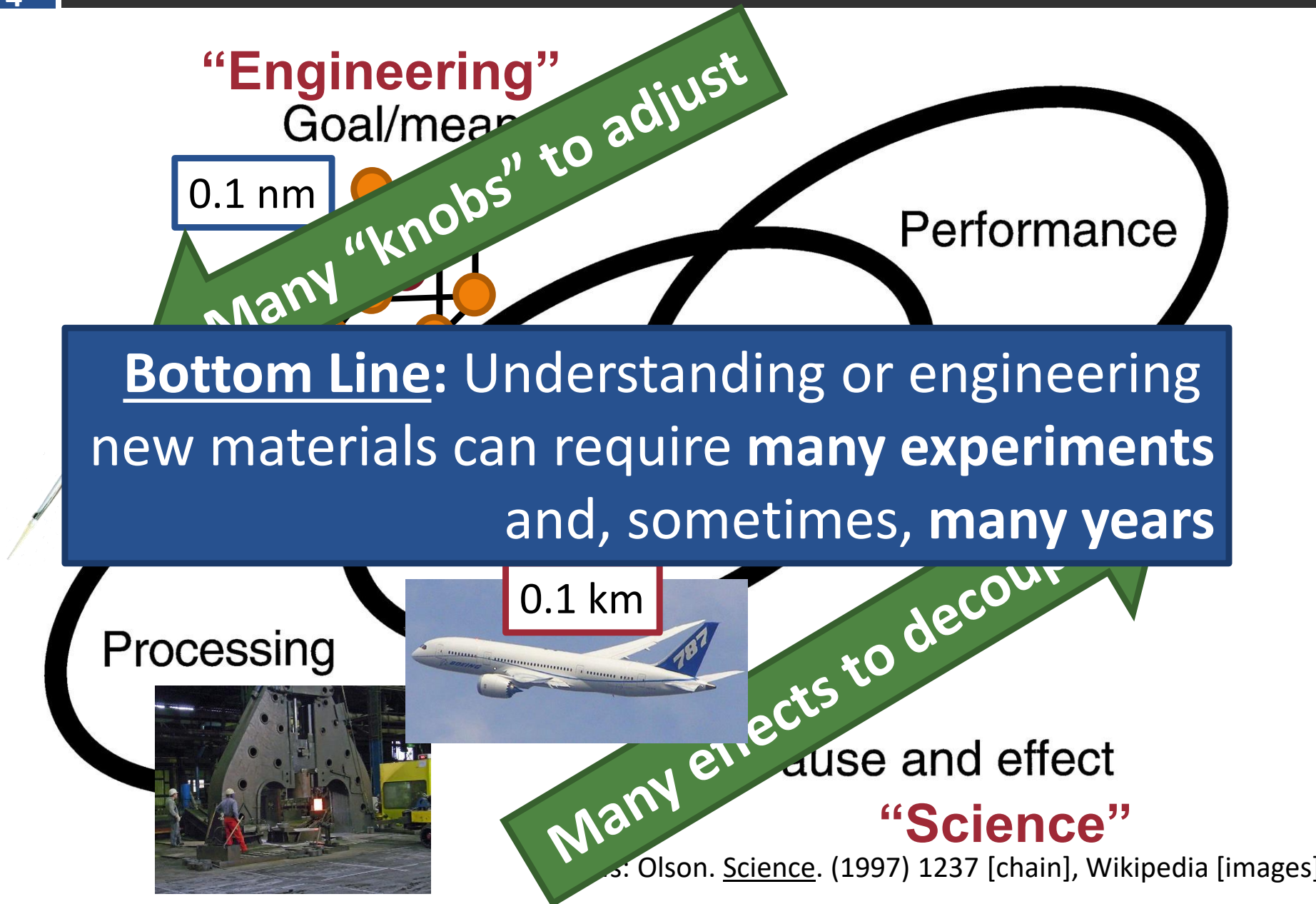
Examples of “Aluminum”

Application				
Key Properties	Conductivity	Low cost Shapeable	Castable Strong	???
Alloy	**Al + Mg**	**Al + Mg,Mn**	**Al + Si,Cu**	???

How do we tailor materials for new technologies?

How can we do this quickly?

Challenges in Engineering Materials



s: Olson. Science. (1997) 1237 [chain], Wikipedia [images]

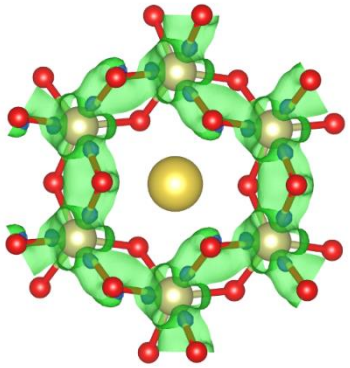
Computational Materials Engineering

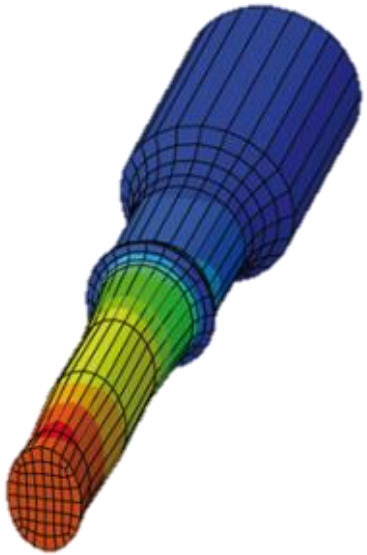
Goal: Accelerate design of materials

Method: Replace experiments with computers

Many Established Tools:

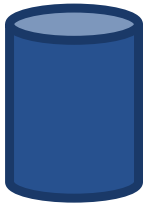
- Density Functional Theory
- Phase Field
- Finite Element Analysis
- Computational Thermodynamics





Emerging Field: **Data-driven Models**

Materials Data



Machine Learning

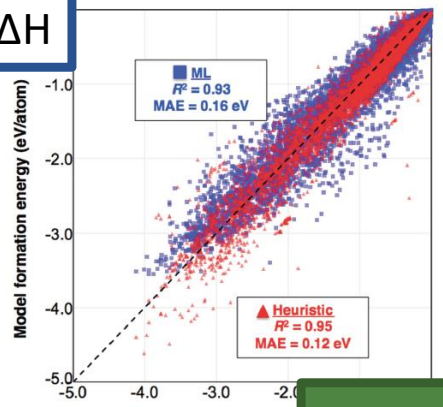
$$\sigma_Y = f(x)$$

Predictive Model

FEA Image: http://www.icams.de/content/research/index.html

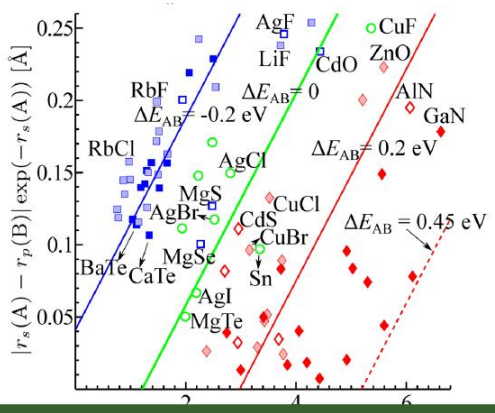
ML + Materials = "Materials Informatics"

ΔH

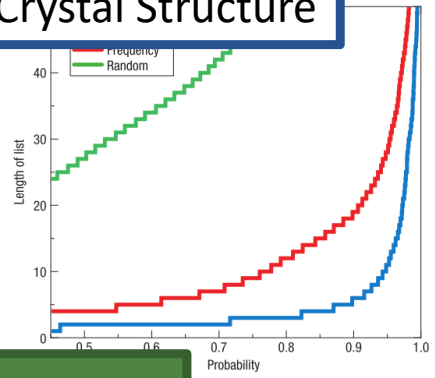


Meredig *et al*. PRB (

ΔH



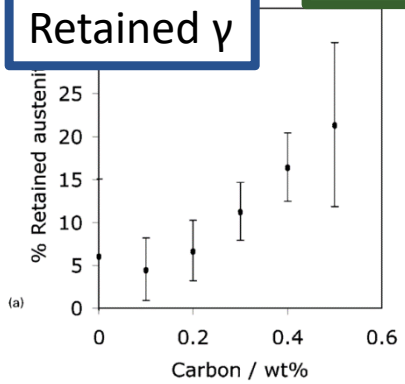
Crystal Structure



Nat. Mat. (2006), 641

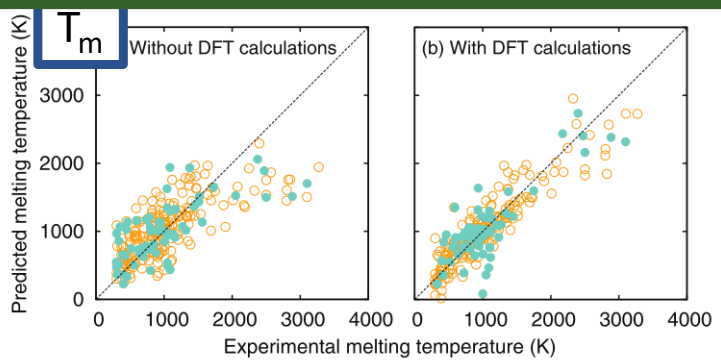
Question for today: Why do this?

Retained γ



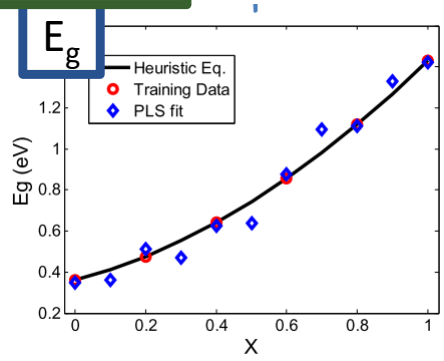
Chatterjee *et al*. MS&T (2007), 819

T_m



Seko *et al*. PRB (2014), 054303

E_g



Srinivasan, Rajan. *Materials* (2013), 279

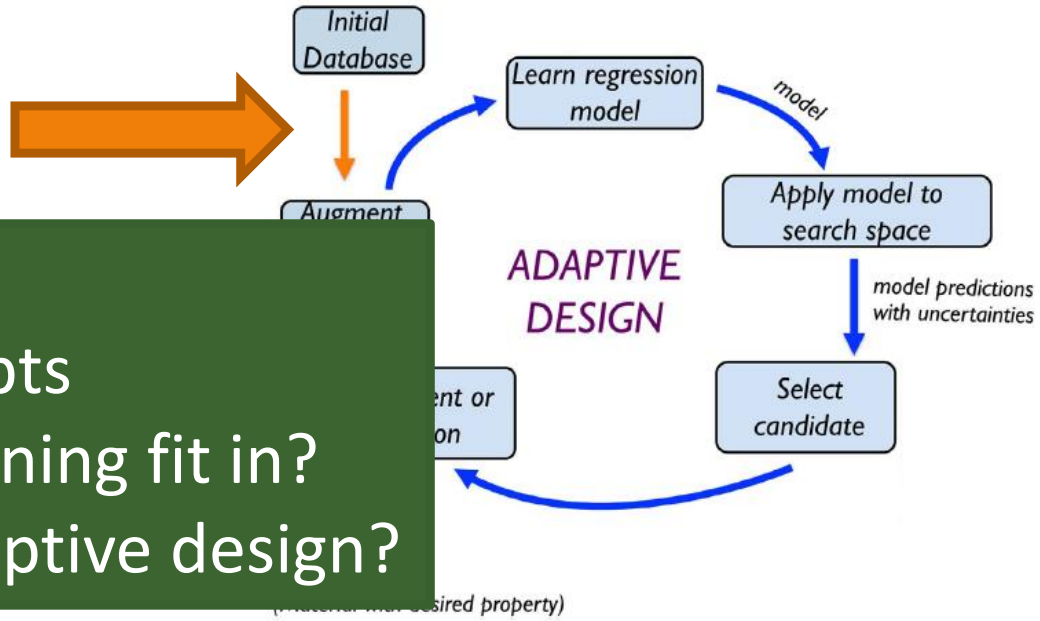
Reviews: Ward, Wolverton. COSSMS. (2017), 167.
Ward et al. MRS Bulletin. (2018), 683.

Why Machine Learning?

Advantages over Physics Models:

✓ Fast	10^4-10^7 evaluations/CPU/sec
✓ Adaptable	Limited need to know underlying physics
✓ Self-correcting	Improves with more data.
✓ Unbiased	Can lead to unexpected predictions

Combined, this means



My Talk Today:

- Dive into these concepts
- Where does deep learning fit in?
- How to do "deep" adaptive design?

Balachandran *et al*. Sci. Rep. (2016), 19660.

ML/DL and Materials Design

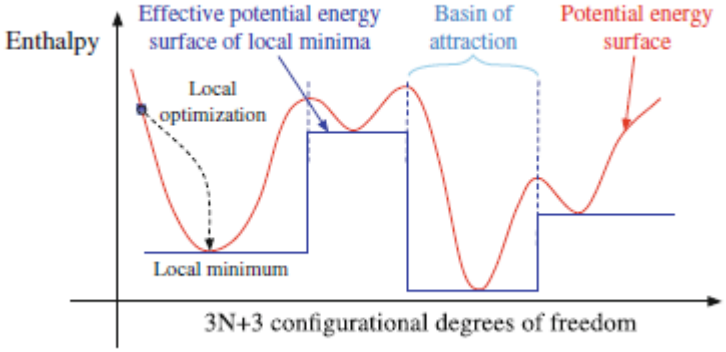
ML models are fast

Physics codes tend to be expensive

Example: Density Functional Theory

DFT: 10^0-10^3 CPU-hr/compound

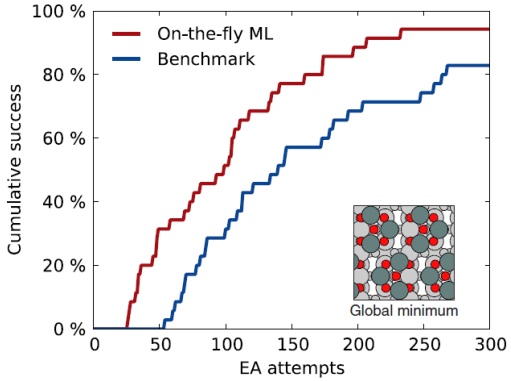
ML: ~10^{-5} CPU-hr/compound



Revard *et al.* Top Curr Chem. (2014), 181–222.

Design requires many evaluations:

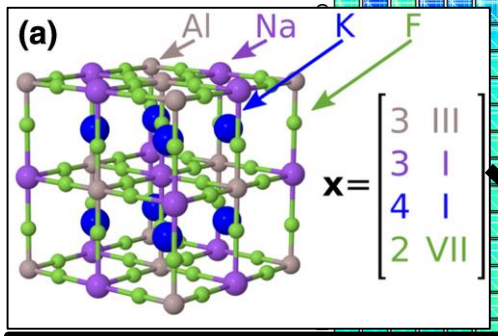
1. High-throughput parameter sweeps
2. GA-based optimization
3. …



Jacobsen et al. PRL (2018), 026102

Machine learning models are useful as fast surrogates

Scanning millions of Elpasolites



Each pixel is a different compound

Found 90 new stable materials, including $NFAl_2Ca_6$, with only 10^4 calculations

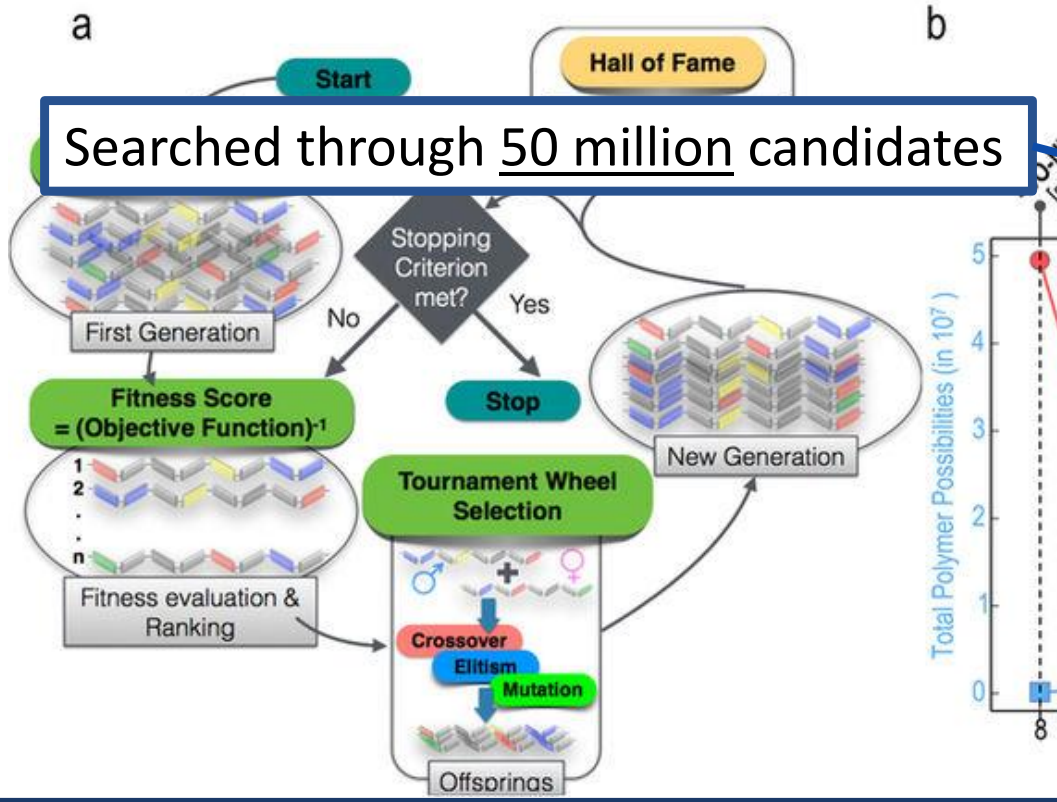
"This outcome directly demonstrates that ML methods can be used for the discovery of stable as well as unconventional chemistries."

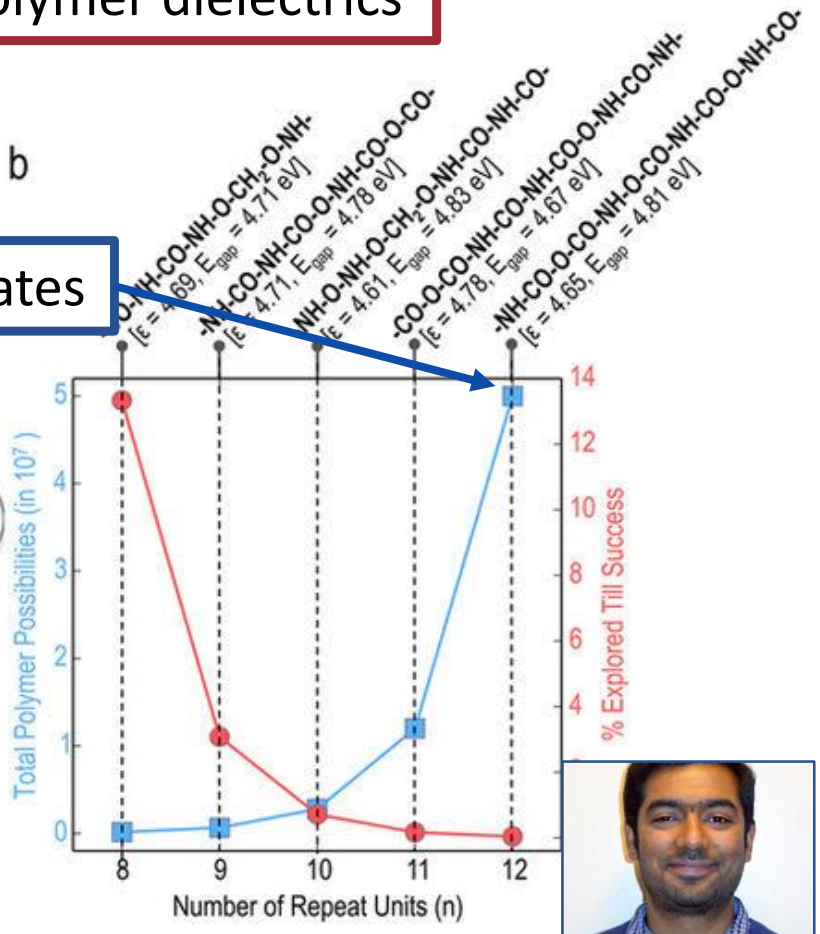
Ref: Faber *et al*. PRL. (2016) 1333502. doi: 10.1103/PhysRevLett.117.135502

ML design of polymer dielectrics

Used ML+Genetic algorithm to find new polymer dielectrics

Searched through 50 million candidates





ML makes otherwise impossible searches practical

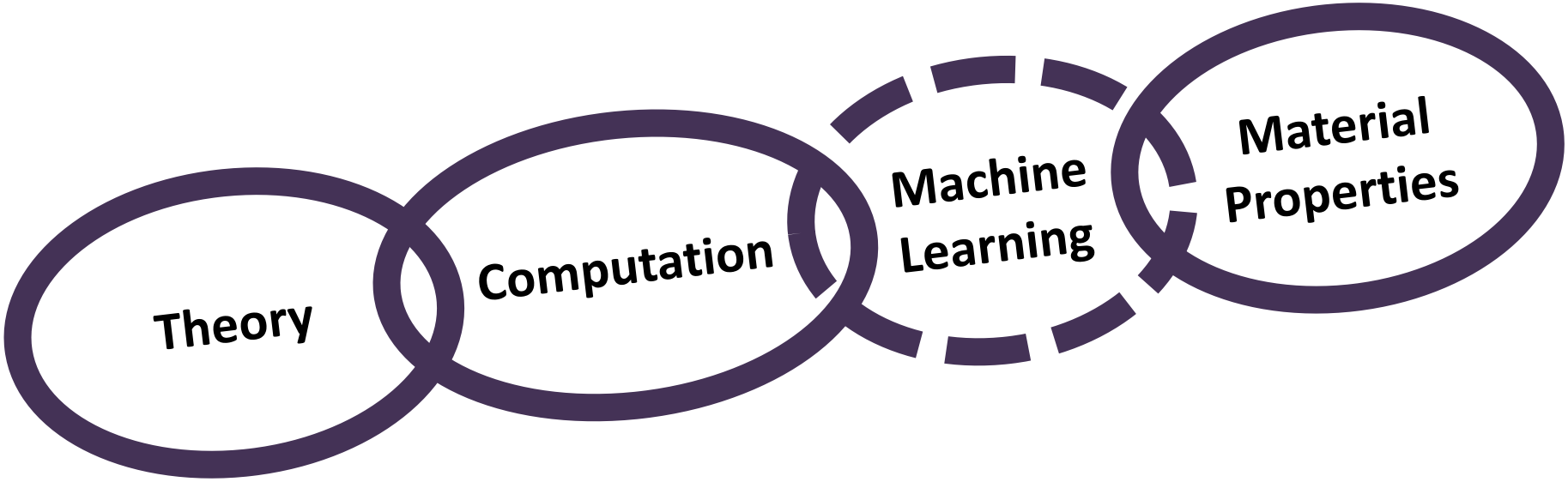
A. Mannodi-Kanakkithodi, CNM

Ref: Mannodi-Kanakkithodi *et al.* Sci Reports. (2016), 20952. doi:10.1038/srep20952

ML models are adaptable

Not all properties are accurately computable (yet!)

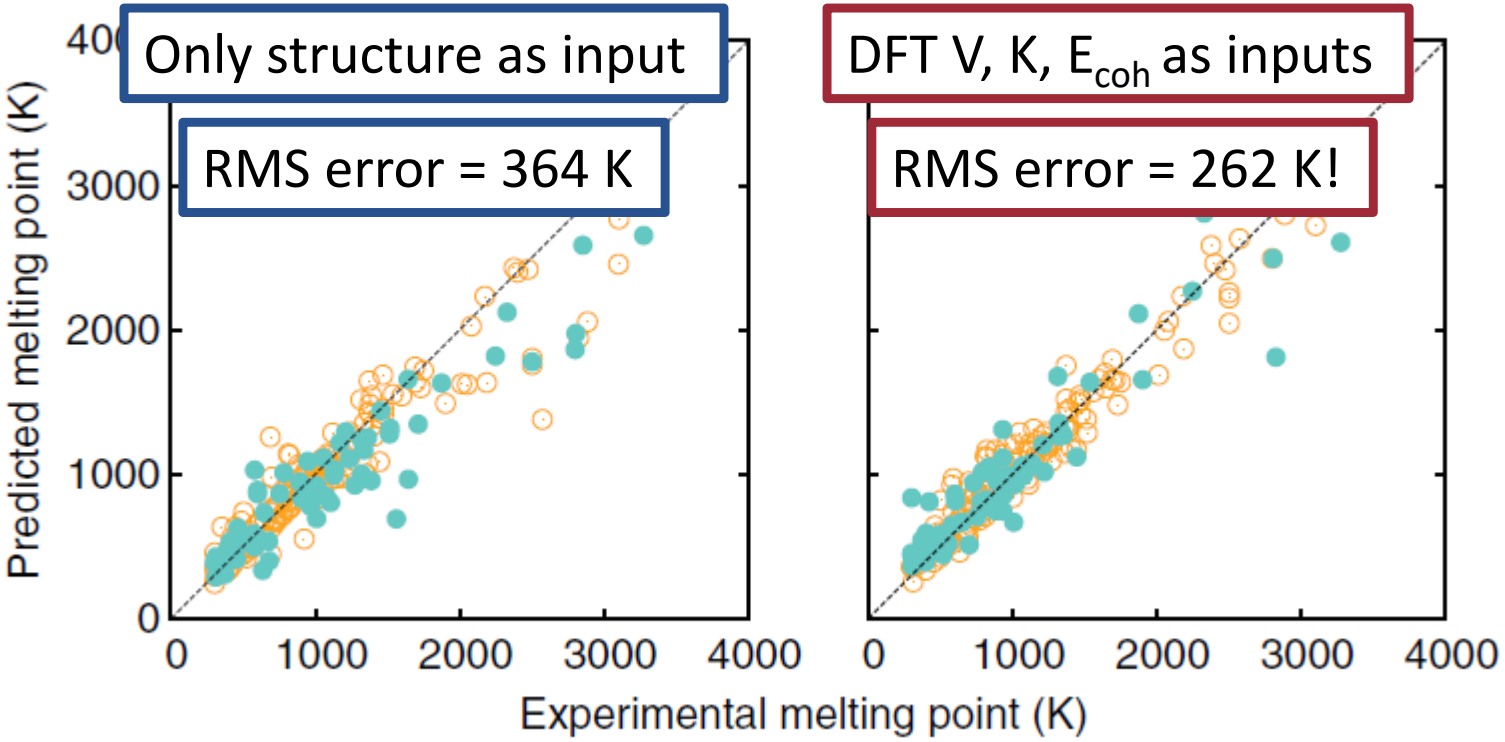
Example: Glass-forming ability of metal alloys



ML can fill in links between computation and properties

Fast melting point calculations

Melting temperature can be computed with DFT, but is very expensive



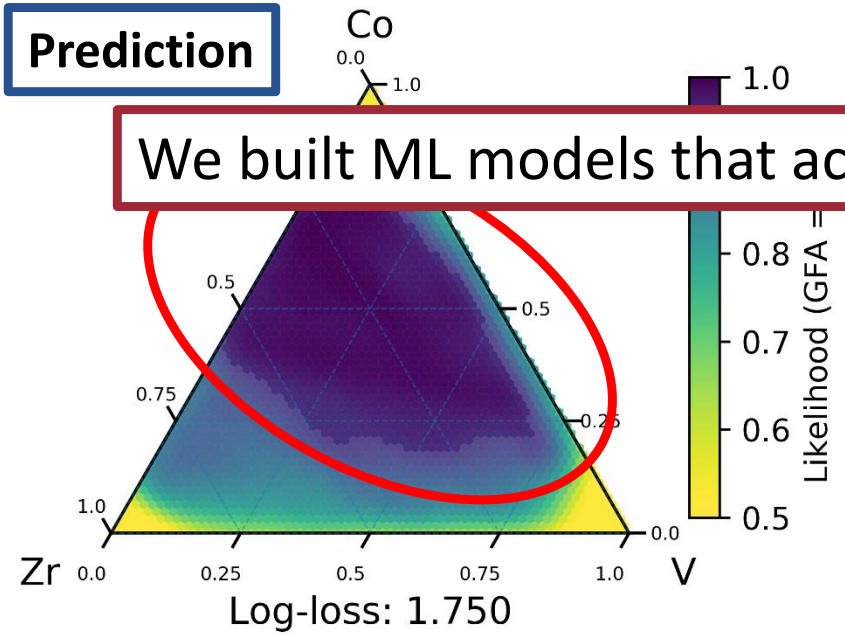
Physics models + ML leads to more accurate, faster predictions

Ref: Seko *et al.*, PRB. (2014), 054303

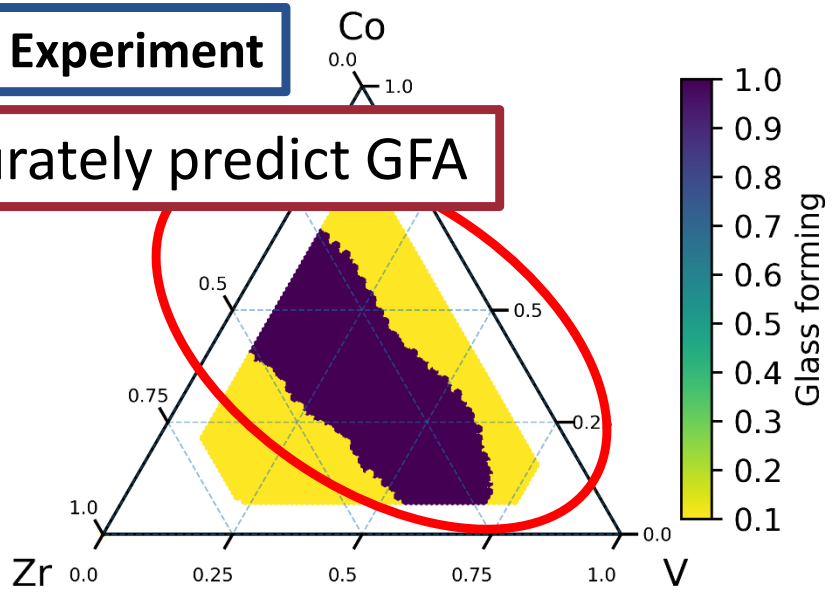
Predicting Glass-forming ability

Few models/theories link composition -> glass-forming ability

Prediction



Experiment



We built ML models that accurately predict GFA

Machine learning fills missing links

J. Hattrick-Simpers, NIST
Fang Ren, SLAC (Apple)
Apurva Mehta, SLAC

Ref: Ren et al. Sci. Adv., eaaq1566

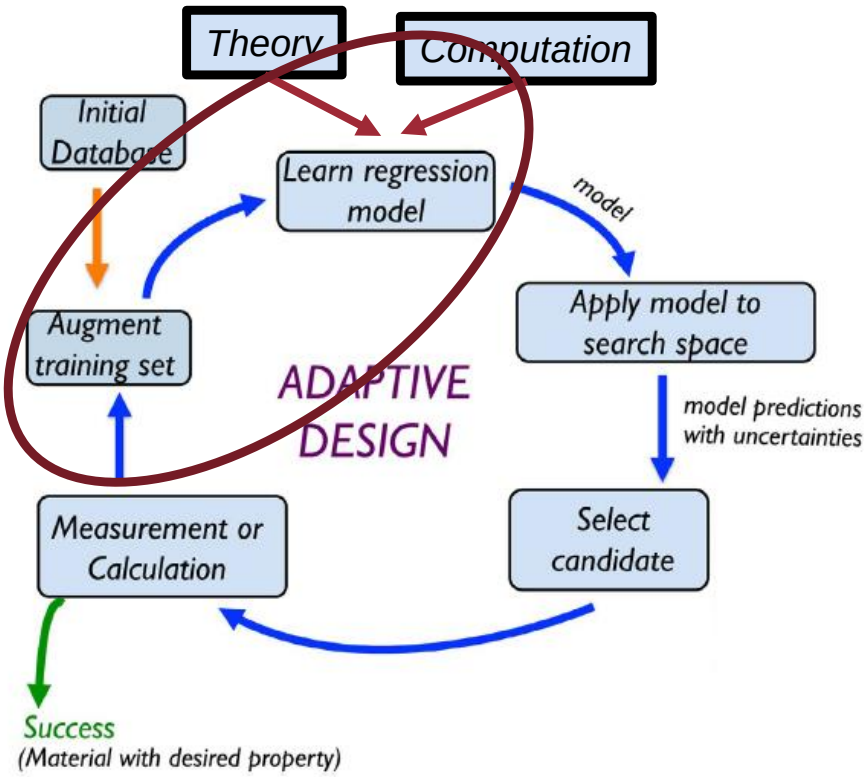
ML models are self-correcting

Machine learning models are trivial to update

Models can [often] be improved by just adding new data

Best case: No human effort

Implication: Updating model predictions faster than making new measurements

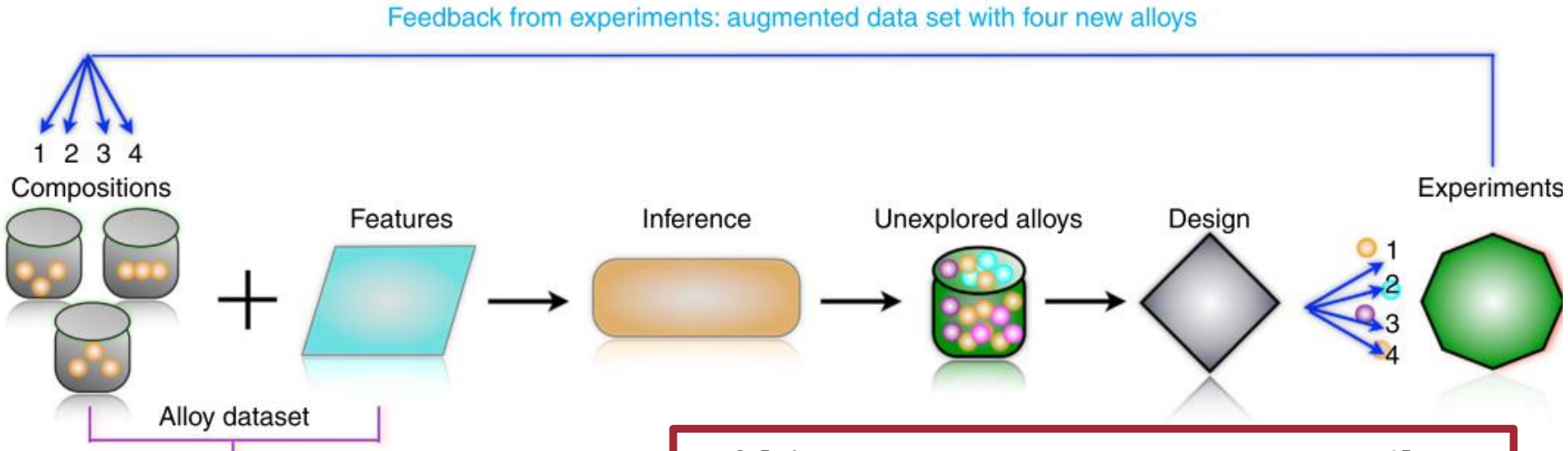


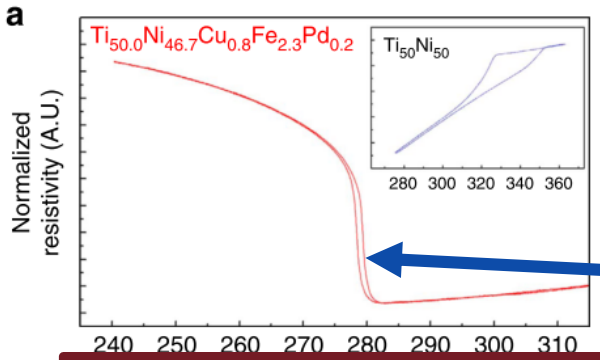
Easy updating allows for adaptive design

Figure: Balachandran *et al*. Sci. Rep. (2016), 19660. doi: 10.1038/srep19660

Adaptive design of shape-memory alloys

Example: Shape memory alloys with small ΔT





Of [the 36 compositions we tested], 14 had smaller ΔT than any of the 22 in the original data set.

Best alloy was 42% lower!

Adaptive search leads to rapid materials design

Ref: Xue *et al*. Nat Comm. (2016), 11241. doi: 10.1038/ncomms11241

What about deep learning?

Why Deep Learning?

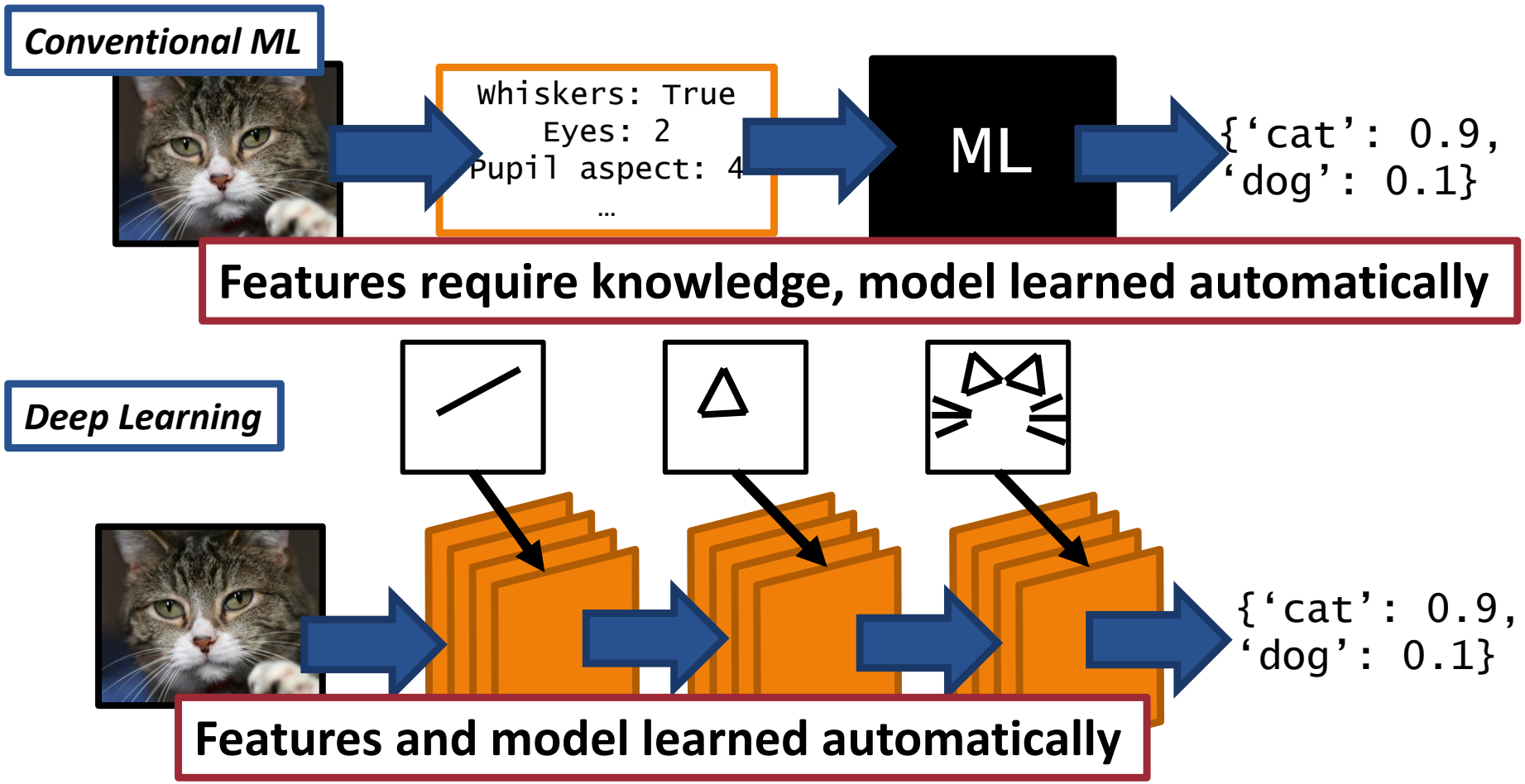


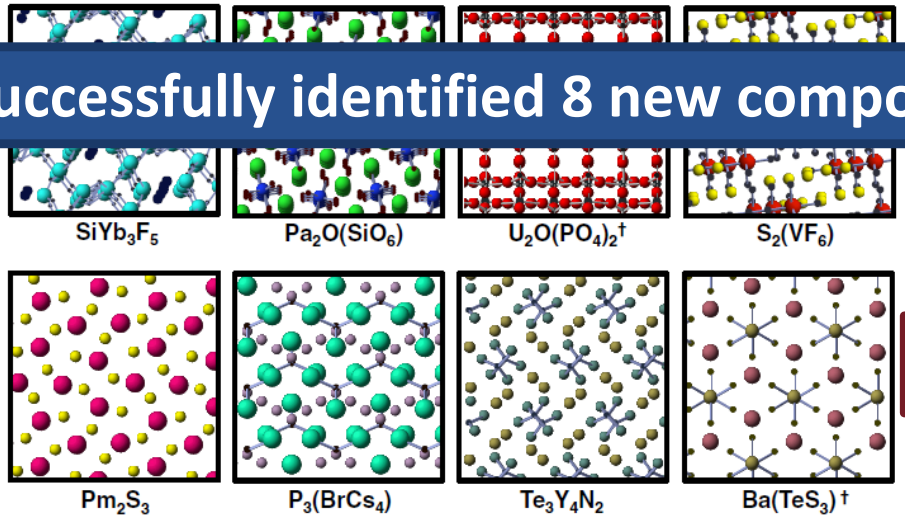
Image: David Corby, 2006

Predicting Crystalline Materials

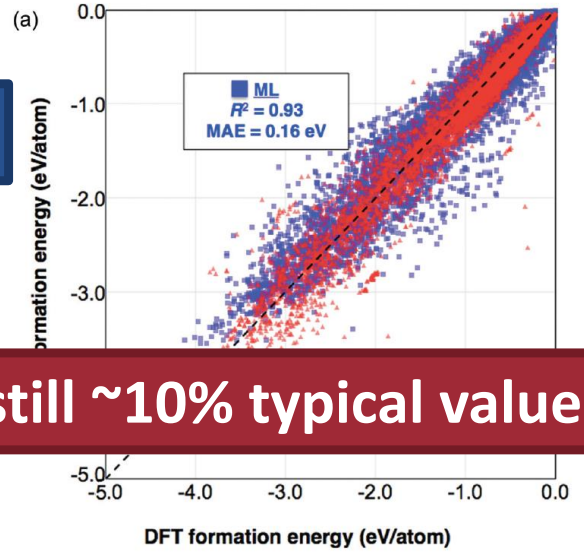
Goal: Identify new crystalline materials

Model: Given, composition predict ΔH_f

Initial attempt: 2014



Successfully identified 8 new compounds!



Error still ~10% typical values

Can we do even better with deep neural networks?

Ref: Meredig *et al*. PRB. (2014), 094104. doi: 10.1103/PhysRevB.89.094104

Training Data

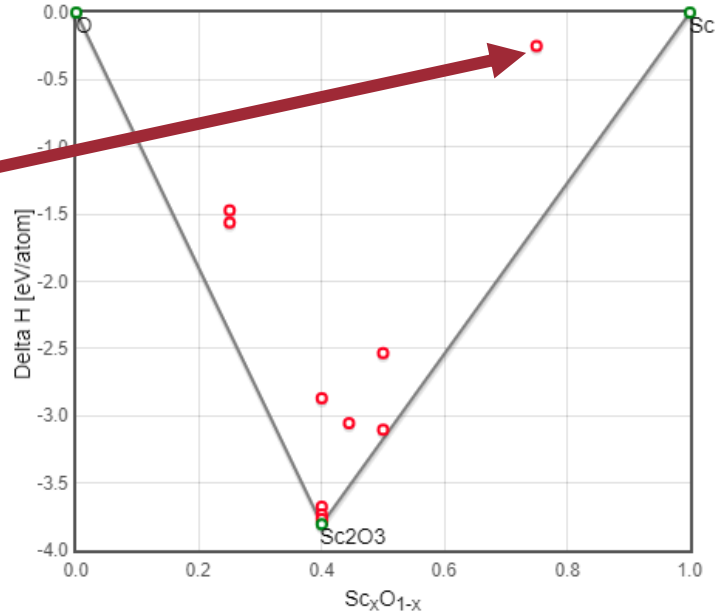
Collect | Process | Represent | Learn

Data Source: Open Quantum Materials Database

~470k DFT Calculations

Why OQMD?

- ✔ Open ✔ Large
- ✔ Contains unstable materials



Training Set:

Subset: Lowest energy structure

Size: 230336

Input: Composition

Output: Formation Enthalpy ΔH_f

Ref: Jha *et al.*, *in review*

“Representation”

Collect → Process → Represent → Learn

Previous state-of-the-art: Compute physical features

Composition

$LiFePO_4$

Element Fractions

x_H	x_{He}	x_{Li}	…
0	0	$1/7$	…

Physical Features

Range Electronegativity: 2.46

Is it charge-neutral: Yes

Maximum T_m: 1538

[…]

With Deep Learning: Rely on representation learning

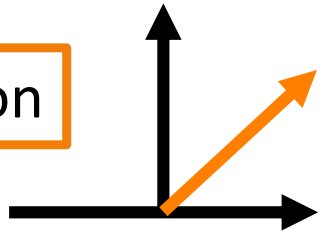
Ref: Jha *et al.*, *in review*

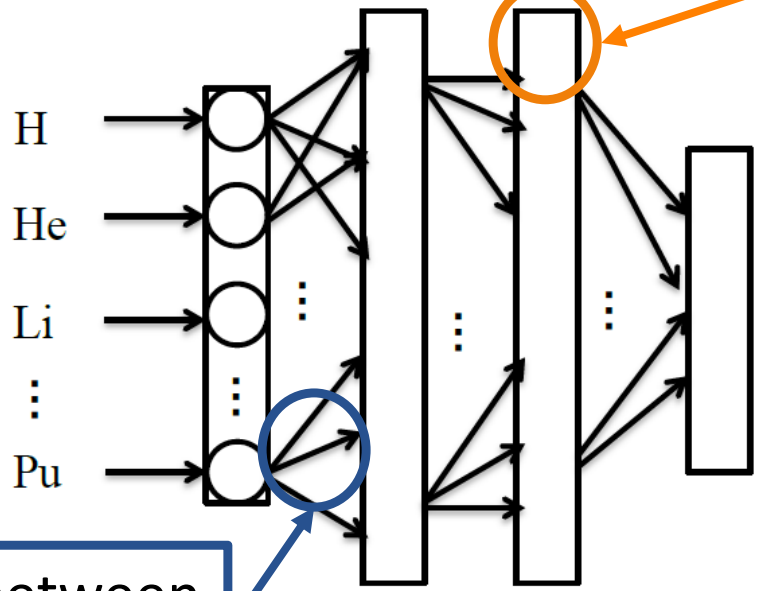
ElemNet: Architecture

Collect > Process > Represent > Learn

Fully connected MLP

ReLU activation





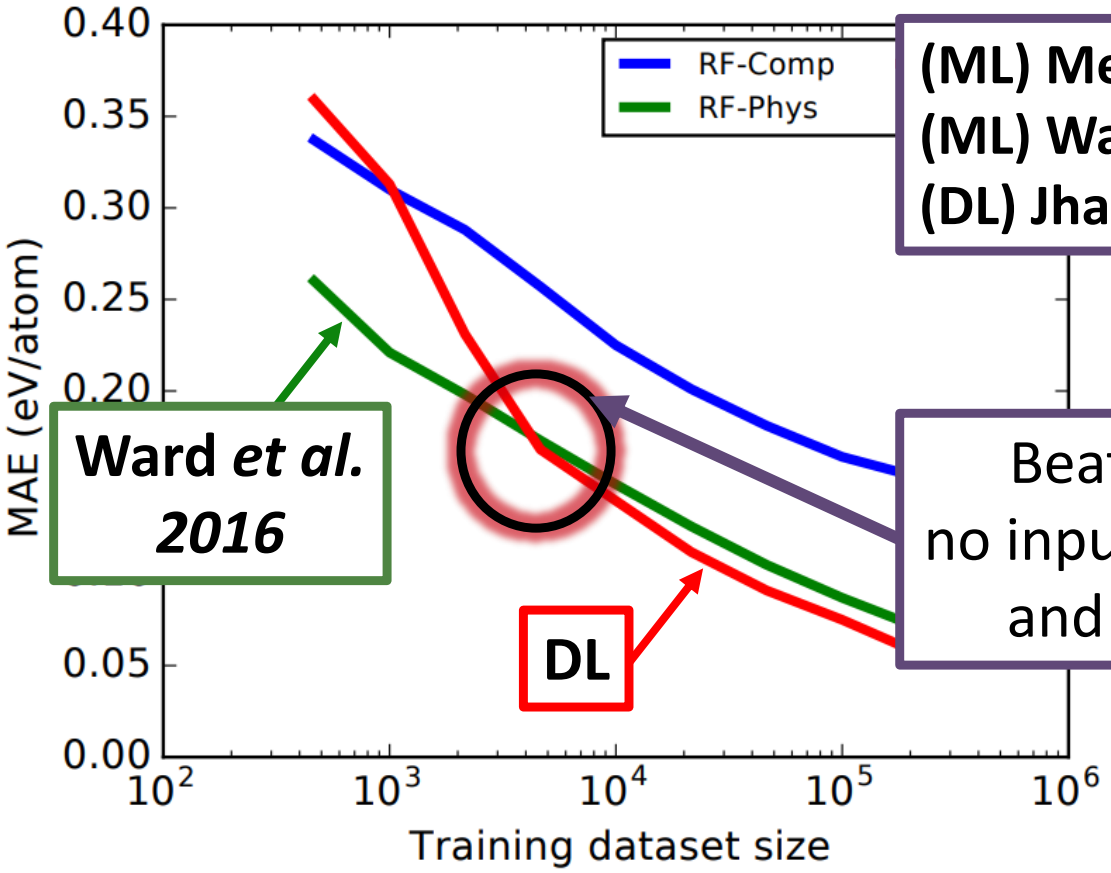
Optimizer: SG
Regularization: L2
Framework: Theano/TF

Dropout between some layers

16 Layers

Ref: Jha *et al.*, *in review*

Better than conventional learning?



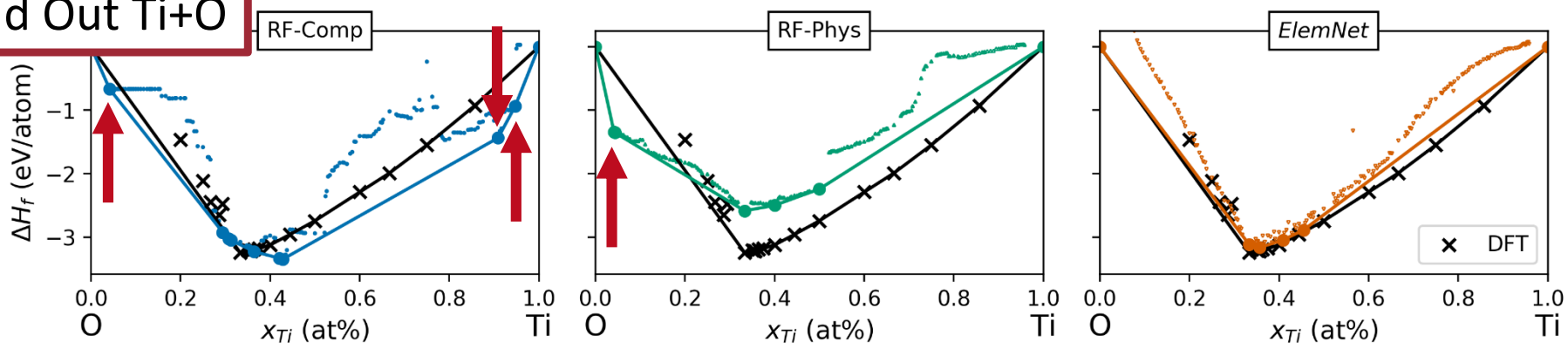
(ML) Meredig 2014: 0.092 eV/atom
(ML) Ward 2016: 0.070 eV/atom
(DL) Jha 2017: 0.055 eV/atom

Beats state-of-the-art with no input from materials scientist, and only 4000 data points!

Ref: Jha *et al.*, *in review*

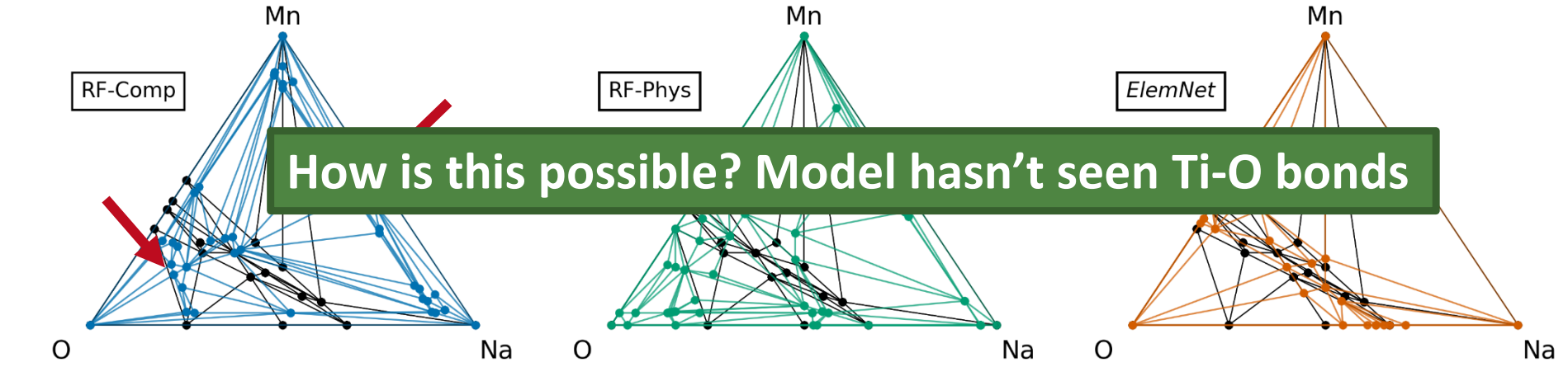
Can DL interpolate between elements?

Hold Out Ti+O



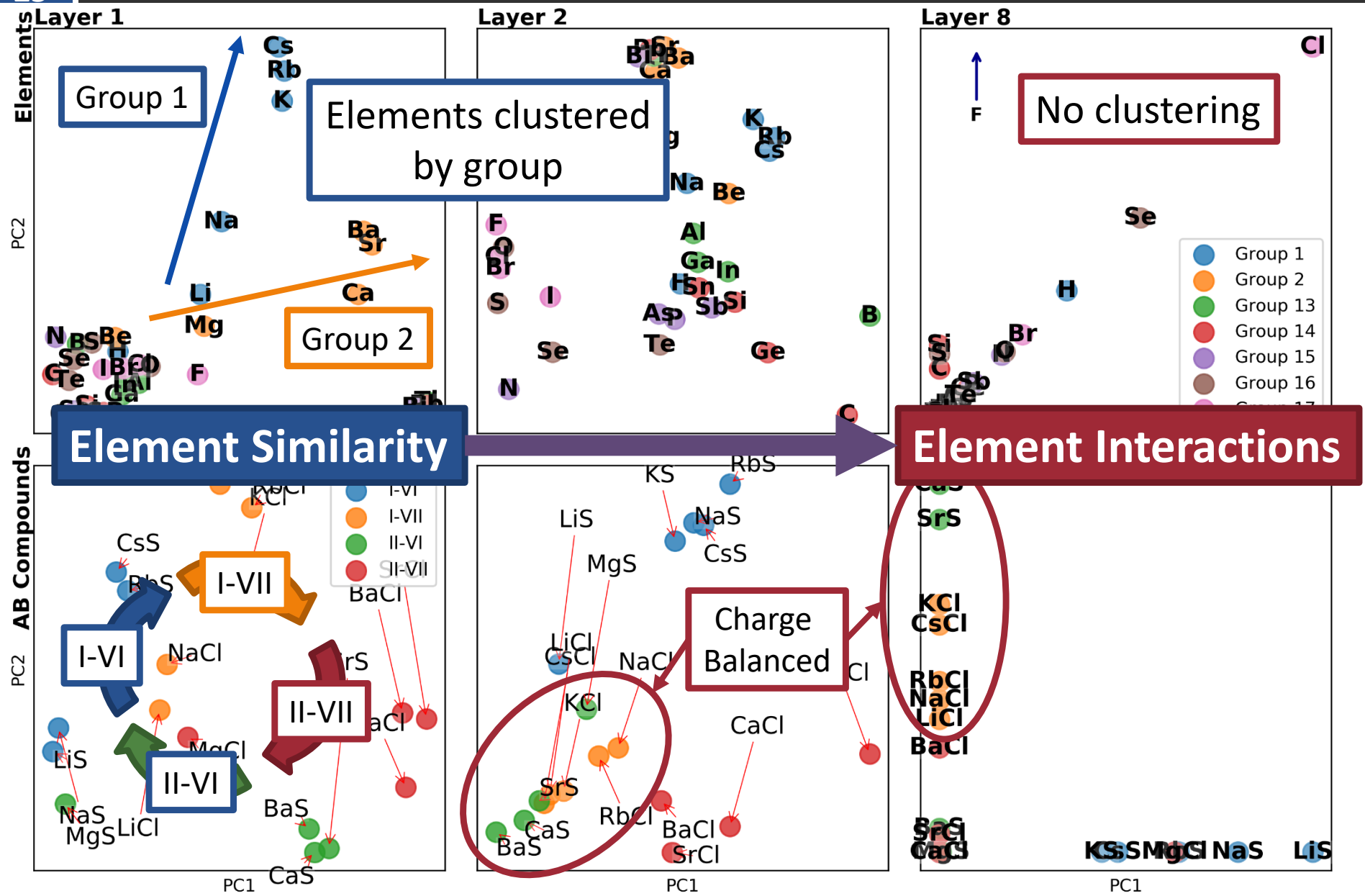
Deep Learning Yields Fewer Spurious Predictions

Hold Out Na-Fe-Mn-O



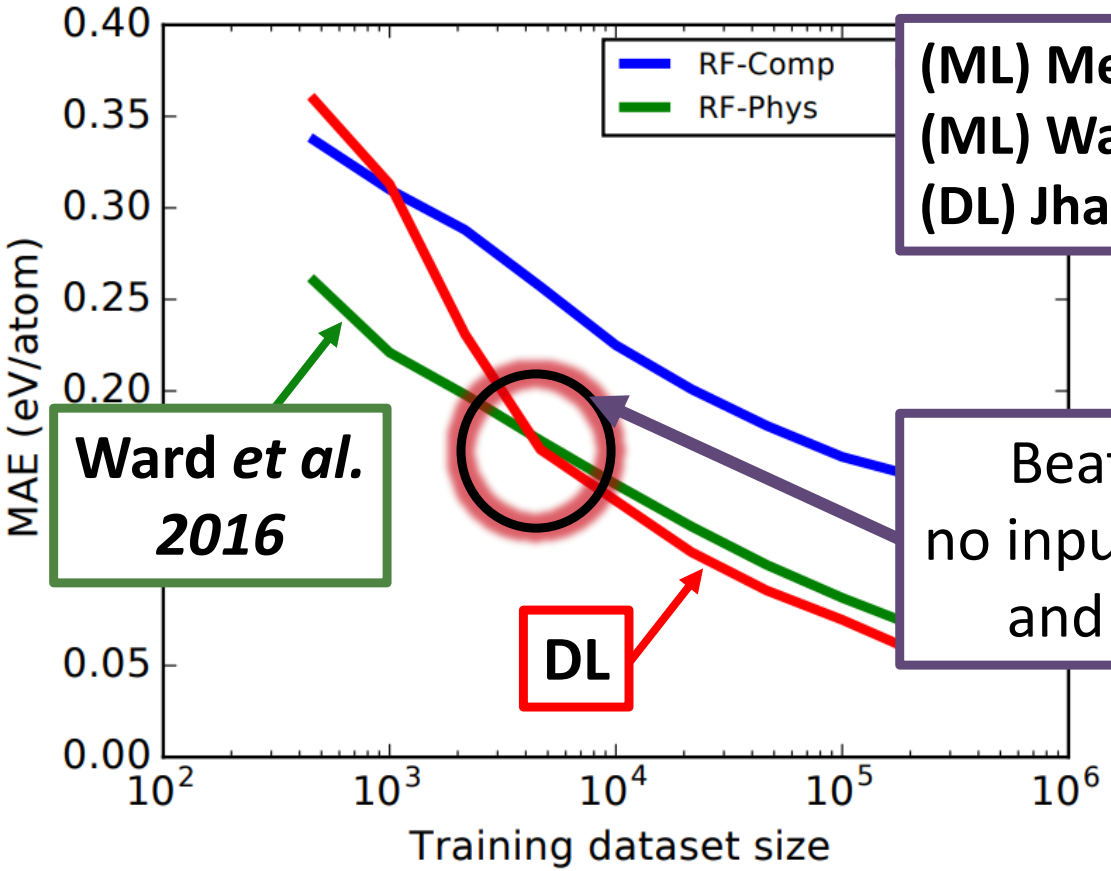
How is this possible? Model hasn't seen Ti-O bonds

How is it working so well?



Example #1: Composition -> Property

Challenge: Given element fractions, predict ΔH_f



(ML) Meredig 2014: 0.092 eV/atom
(ML) Ward 2016: 0.070 eV/atom
(DL) Jha 2017: 0.055 eV/atom

Ward *et al.* 2016

DL

Beats state-of-the-art with no input from materials scientist, and only 4000 data points!

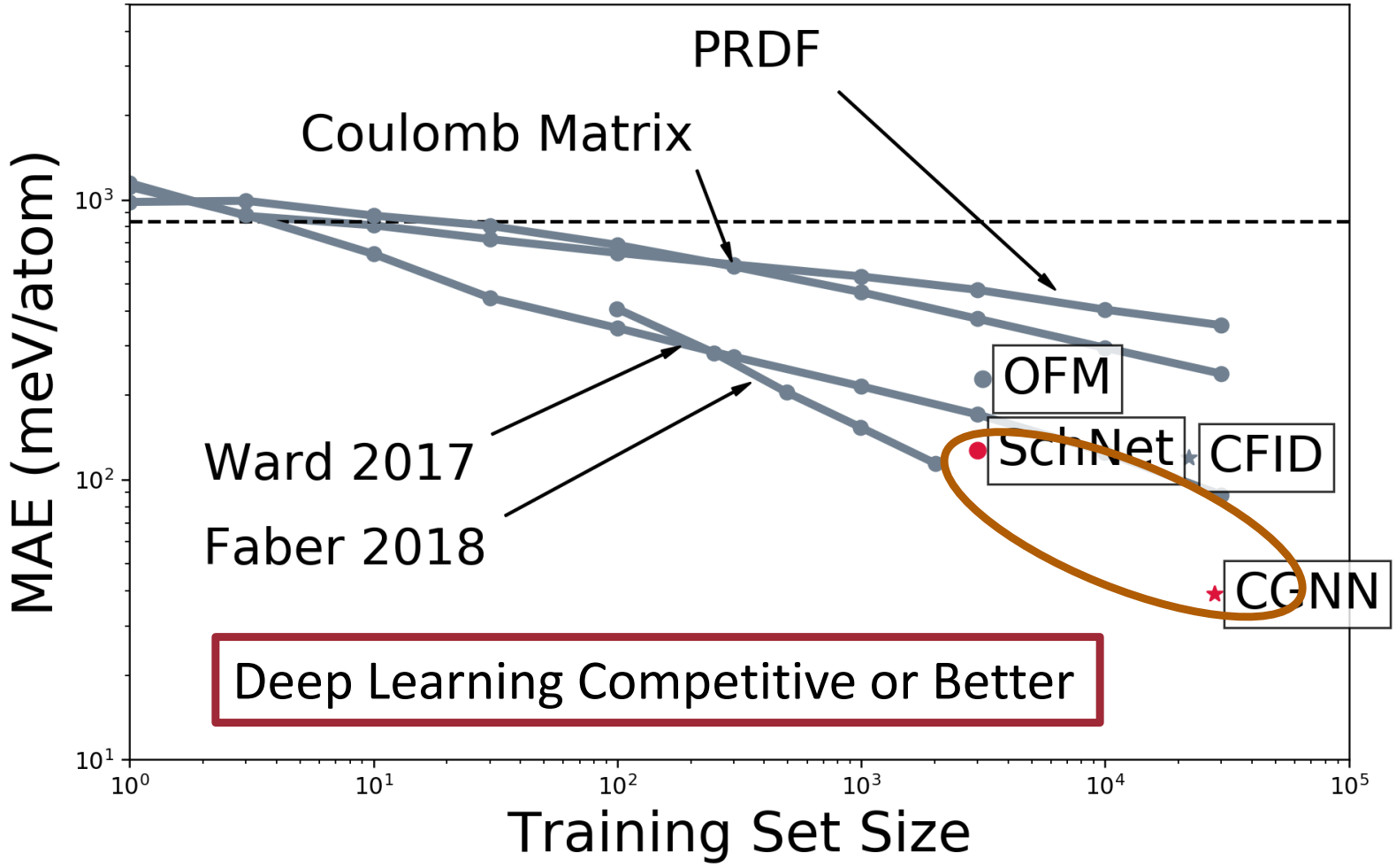
Deep learning could change the way we approach problems

Ref: Jha *et al.*, *in preparation*

Example #2: Crystal Structure

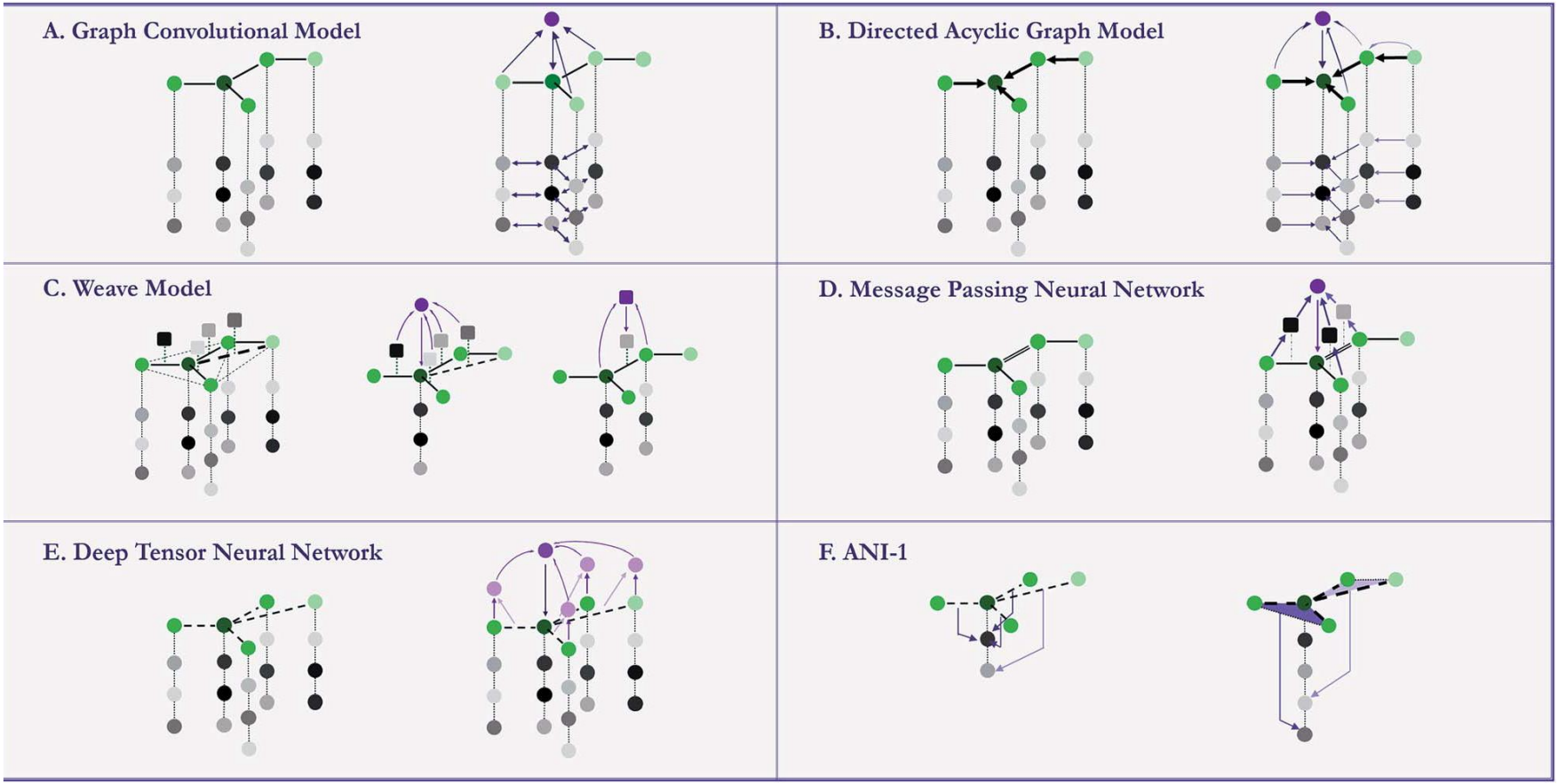
Dataset: 32k DFT ΔH_f from the OQMD

Test: Remove 1000, train on *N* remaining



CM: Faber *et al*. Int J Quantum Chemisty. 2015; PRDF: Schutt *et al.* PRB. (2014)

Example #3: Molecules



Many methods available for molecular data!

Ref: Wu *et al.* Chem Sci: 10.1039/C7SC02664A

DL not a cure-all, but taking over

Table 3: Summary of performances(test subset): conventional methods versus graph-based methods. Graph-based models outperform conventional methods on 11/17 datasets.

Category	Dataset	Metric	Best performances - conventional methods	Best performances - graph-based methods
Quantum Mechanics	QM7	MAE	KRR(CM): 10.22	**DTNN: 8.75**
	QM7b	MAE	KRR(CM): 1.05	**DTNN: 1.77***
			ask: 0.0150	**MPNN: 0.0143**
			ask(CM): 4.35	**DTNN: 2.35**
			ost: 0.99	**MPNN: 0.58**
			ost: 1.74	**MPNN: 1.15**
			ost: 0.799	**GC: 0.655**
			: 0.129	**GC: 0.136**
			Multitask: 0.184	Weave: 0.109
Biophysics	HIV	AUC-ROC	**KernelSVM: 0.792**	GC: 0.763
	BACE	AUC-ROC	**RF: 0.867**	Weave: 0.806
	PDBbind(full)	RMSE	**RF(grid): 1.25**	GC: 1.44
Physiology	BBBP	AUC-ROC	**KernelSVM: 0.729**	GC: 0.690
	Tox21	AUC-ROC	KernelSVM: 0.822	**GC: 0.829**
	ToxCast	AUC-ROC	Multitask: 0.702	**Weave: 0.742**
	SIDER	AUC-ROC	**RF: 0.684**	GC: 0.638
	ClinTox	AUC-ROC	Bypass: 0.827	**Weave: 0.832**

* As discussed in section 4.4, DTNN outperforms KRR(CM) on 14/16 tasks in QM7b while the mean-MAE is skewed due to different magnitudes of labels.

For benchmark problems of learning from molecular data, conventional ML better for **6/17** cases

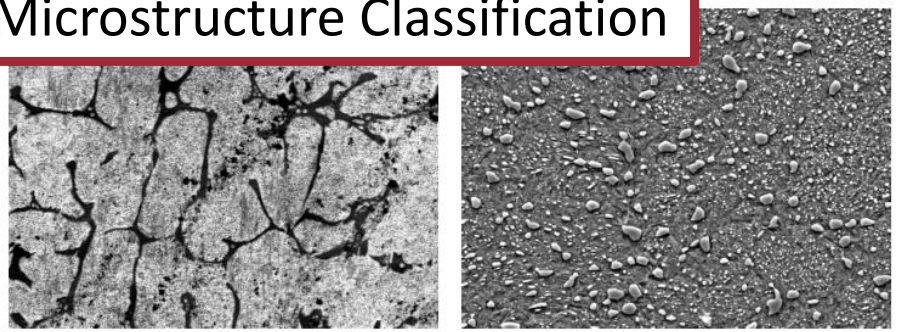
Ref: Wu *et al.* Chem Sci: 10.1039/C7SC02664A

Example 4: Microstructural Data

Major focus area: DL and microstructural data

Why? DL techniques work well with images

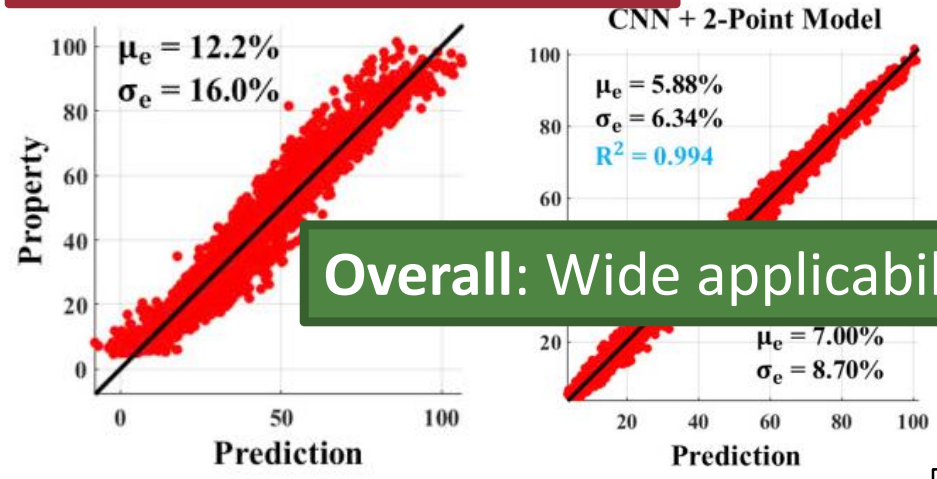
Microstructure Classification



(a) Network (b) Spheroidite

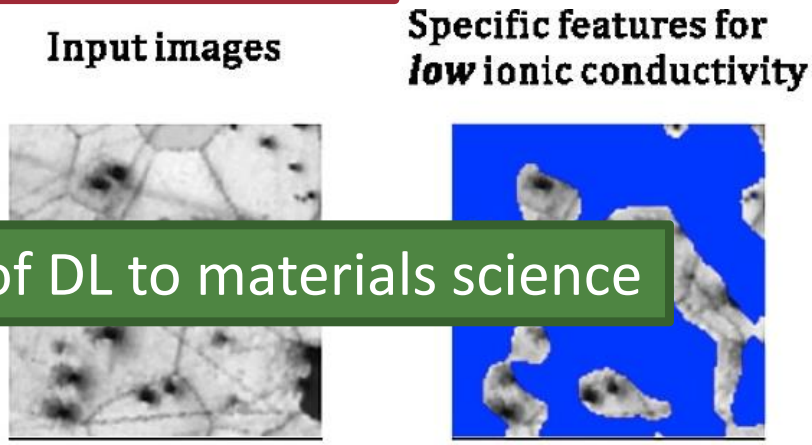
Ling *et al*. arXiv: 1711.00404v1

Surrogate Models for FEM



Cecen *et al*. Acta Mat. (2018), 76

Ionic Conductivity



Kondo *et al*. Acta Mat. (2017), 29

Overall: Wide applicability of DL to materials science

Summary

Why use machine learning in materials?

✔ Fast ✔ Adaptable ✔ Self-correcting ✔ Unbiased

Where does deep learning fit in?

- More accurate than conventional ML
  - For many types of data!

Where's the big opportunity(s)?

- Adaptive design for materials
- Integrating physics into deep learning
- Integrating deep learning into design

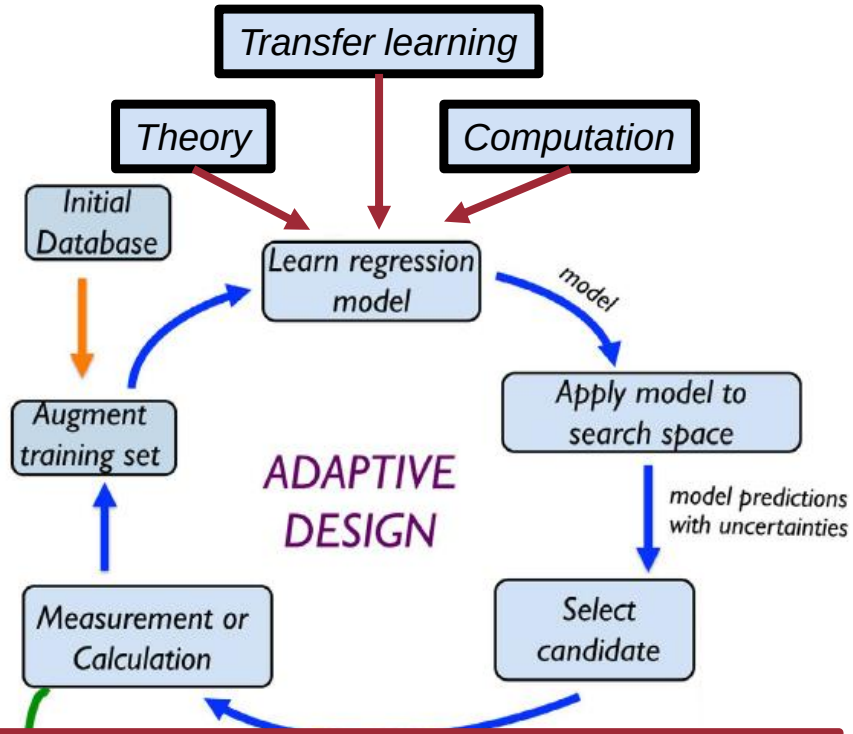
Goal: Tight integration with design

Future: *Adaptive design with automated experiments*

What do we already have?:

- Materials data
- ✓ Representations
- ✓ ML algorithms

Need: Tight integration with materials design



Problem: Disconnect between method developers and model users

Figure: Balachandran *et al*. Sci. Rep. (2016), 19660. doi: 10.1038/srep19660

Argonne NATIONAL LABORATORY

ARGONNE LEADERSHIP COMPUTING FACILITY

U.S. DEPARTMENT OF ENERGY

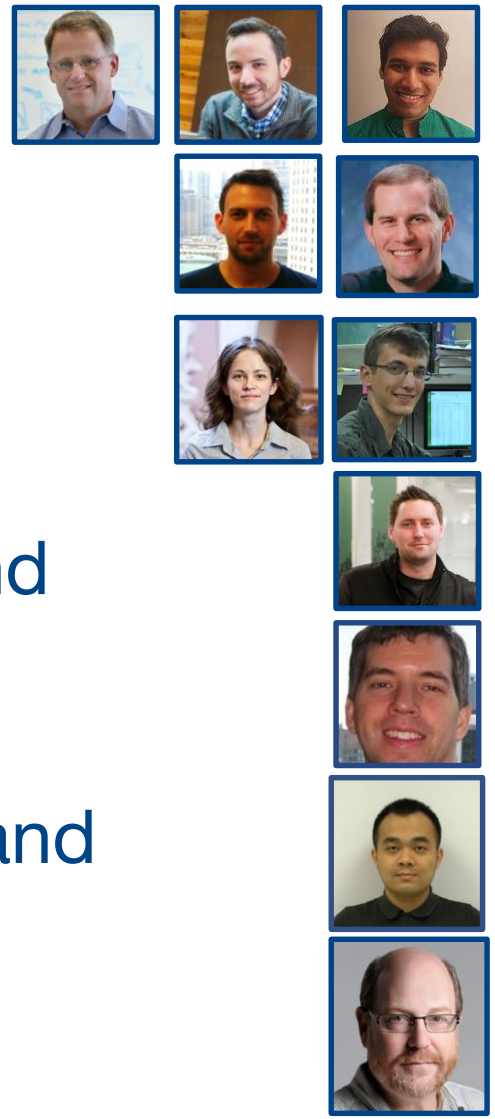
DLHub

Data and Learning Hub for Science

Ben Blaiszik (bblaiszik@anl.gov), Ryan Chard, Logan Ward,
Kyle Chard, Zhuozhao Li, Anna Woodard, Yadu Babuji,
Steve Tuecke, Mike Franklin, Ian Foster

Funding: 2018 Argonne Advanced Computing LDRD

Data and Learning Hub for Science (DLHub)



- Collect, publish, categorize models from many disciplines (materials science, physics, chemistry, genomics, etc.)
- Serve models via API to simplify sharing, consumption, and access
- Enable new science through reuse, real-time integration, and synthesis of existing models

Funding: 2018 Argonne Adv. Computing LDRD

Argonne NATIONAL LABORATORY

ARGONNE LEADERSHIP COMPUTING FACILITY

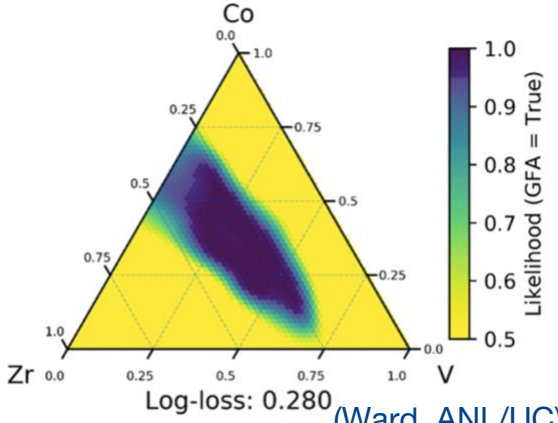
PETREL

parsl

Select DLHub Use Cases

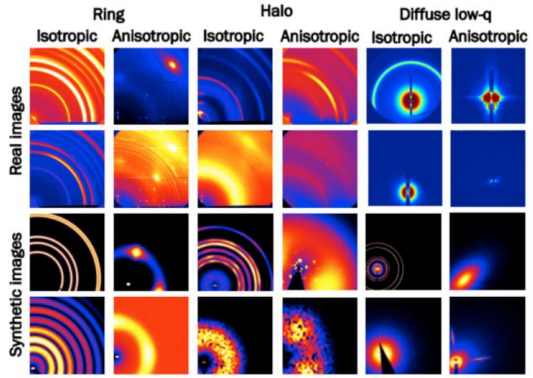
Model-driven Experimentation and Data Tagging

- Metallic glass discovery [active learning]



(Ward, ANL/UC)

- XRD beamline image tagging

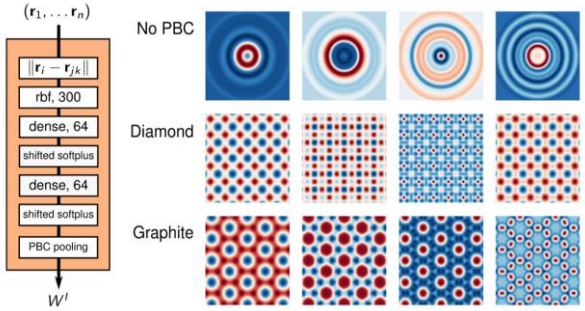


	probability	tags
1	0.999780	Beam Off Image
6	0.745707	Halo
13	0.364863	Strong scattering
7	0.082574	High background
12	0.027826	Ring
8	0.012137	Higher orders
15	0.008106	Weak scattering
9	0.001964	Linear beamstop
16	0.001744	Wedge beamstop
2	0.000982	Circular Beamstop

(Yager, BNL)

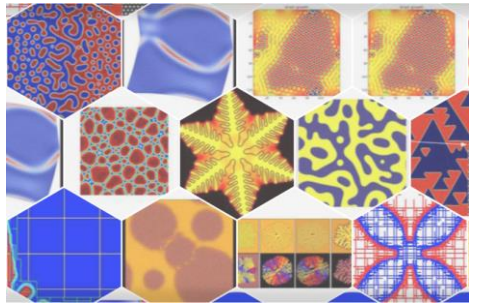
Community Model Benchmarking

- Crystal structure



(Ward, ANL/UC)

- NIST PFHub



(Wheeler, Warren, Heinonen NIST/UC/Argonne/NU)

Automated Model Retraining with New Data

- Models linked to dynamic data sources

(Center for Hierarchical Materials Design NIST/UC/Argonne/NU)

Funding: 2018 Argonne Adv. Computing LDRD

Argonne NATIONAL LABORATORY

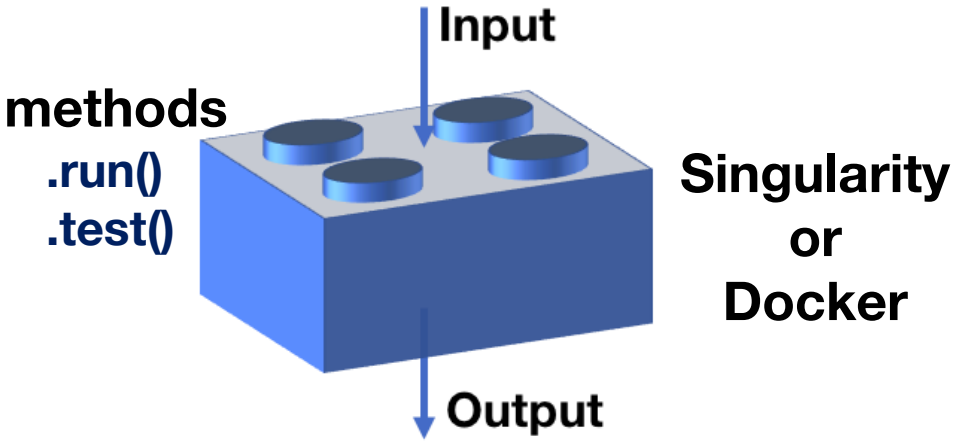
ARGONNE LEADERSHIP COMPUTING FACILITY

PETREL

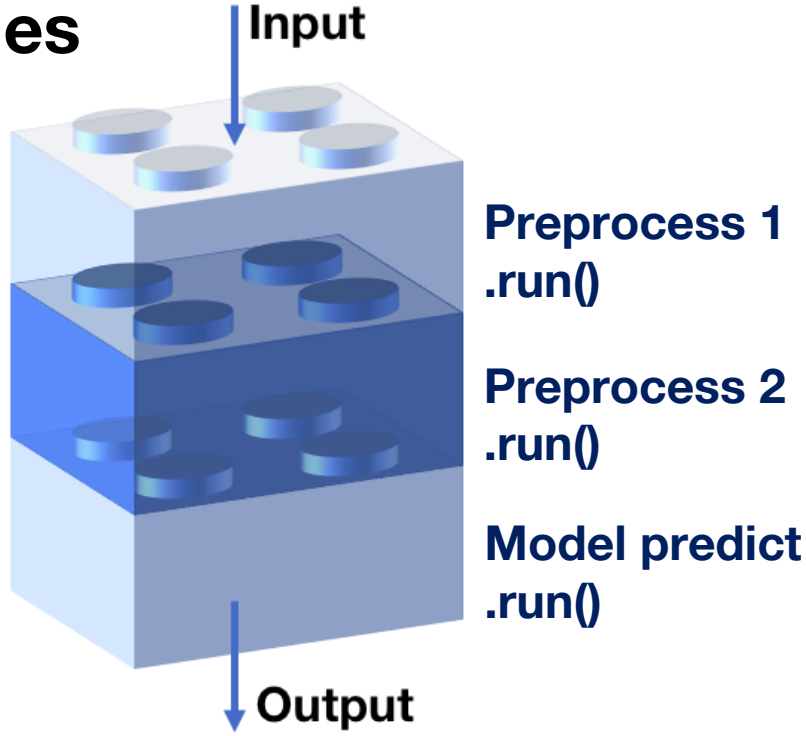
parsl

DLHub Servables and Pipelines

Servables



Pipelines



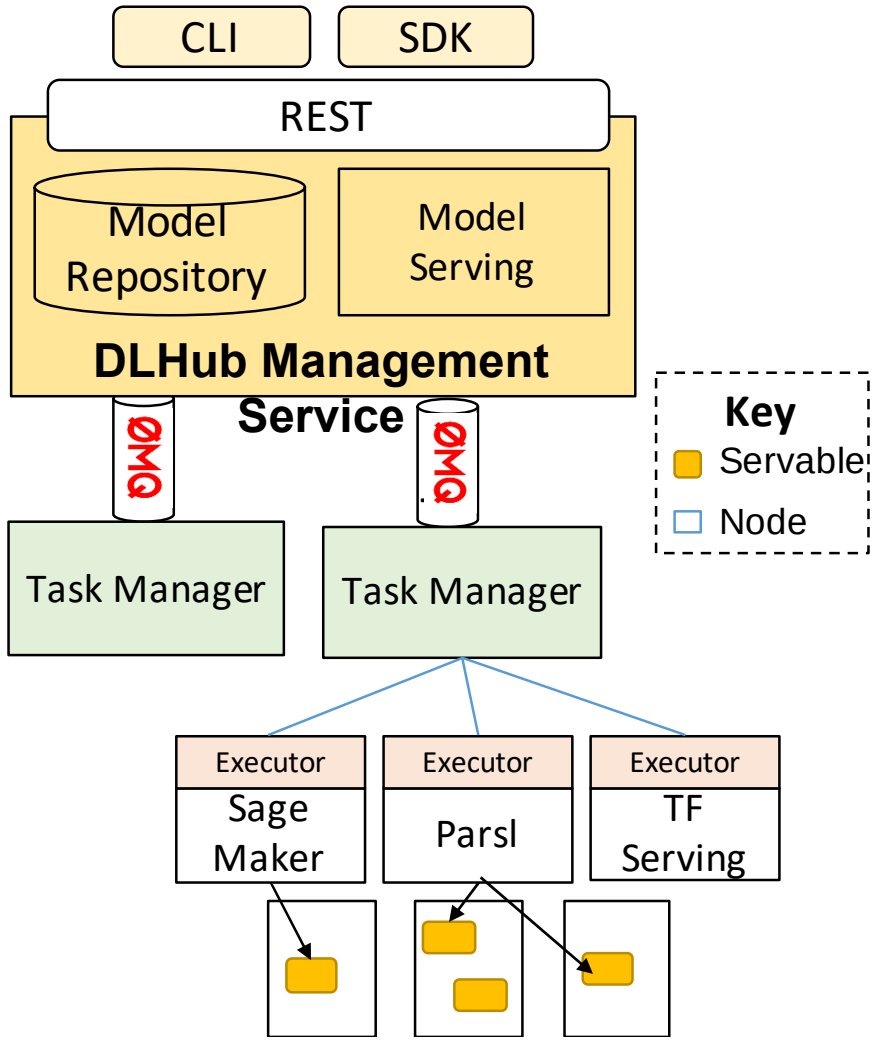
Funding: 2018 Argonne Adv. Computing LDRD

Argonne NATIONAL LABORATORY

ARGONNE LEADERSHIP COMPUTING FACILITY PETREL parsl

DLHub Architecture

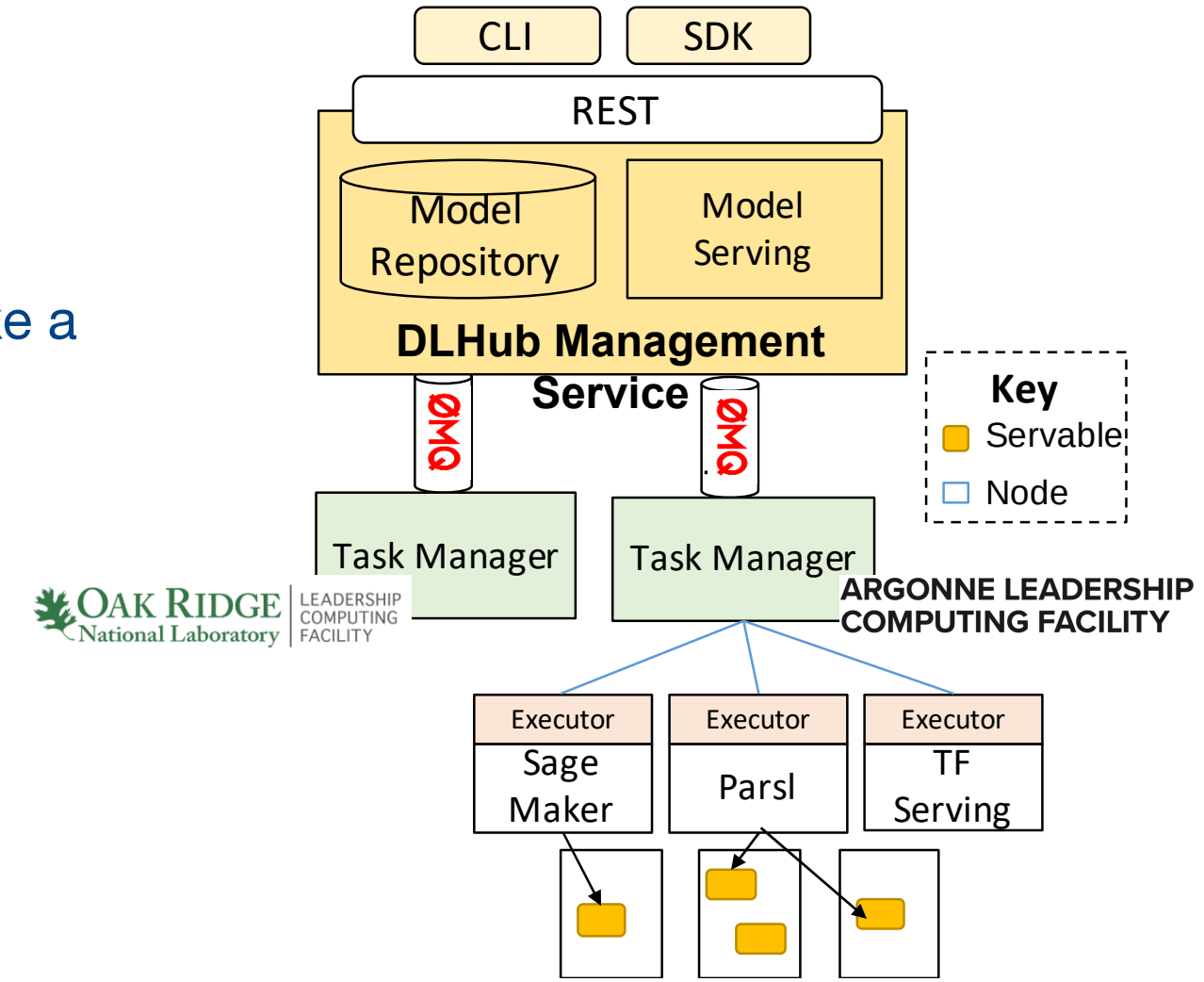
- REST API with Python SDK (available) / CLI (delivery in Nov. 2018)
  - Support model markup, data staging, registration, and invocation
- Model Repository
  - Container registry
  - Advanced search functions
  - Identifier minting capabilities



https://github.com/DLHub-Argonne

DLHub Architecture

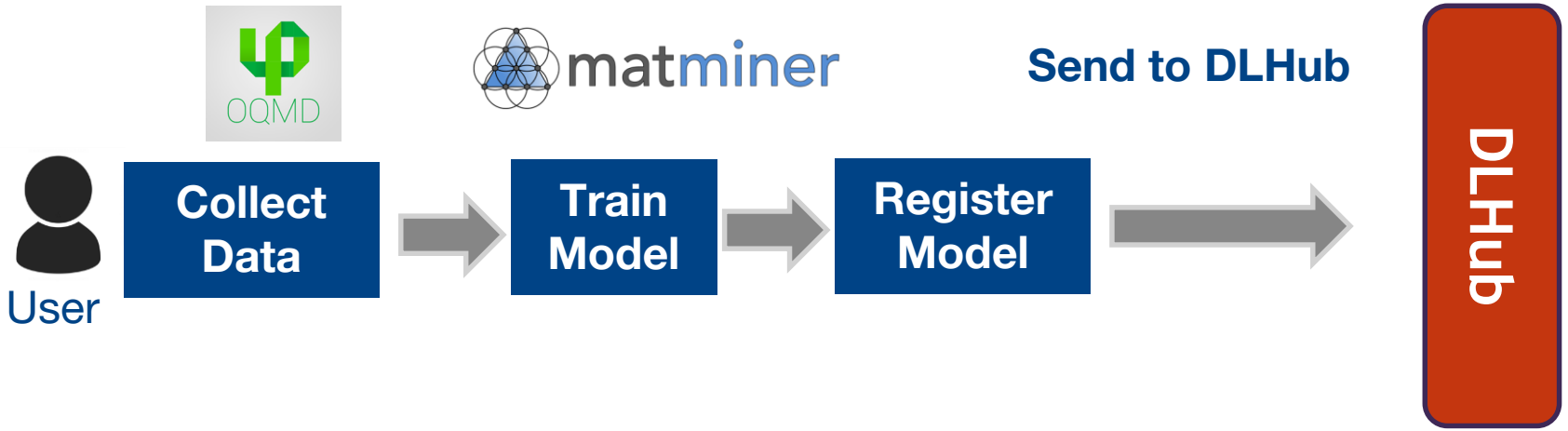
- Task Managers (TM) to support execution on various compute resources
- Executors chosen by TM to invoke a given servable'
- Caching at TM
- Data staging with Globus
- Batch submissions
- Scalability through deployment of model replicas



DLHub Model Registration and Publication

- **Register** model metadata, weights, and files to improve discoverability and reusability
- **Containerize** model to enhance interoperability
- **Identify** model with a permanent identifier (e.g., DOI, minid)
- **Version** model and data pre/post processing steps
- **Deploy** model with simplified interfaces for users
- **Control** access to model metadata and usage

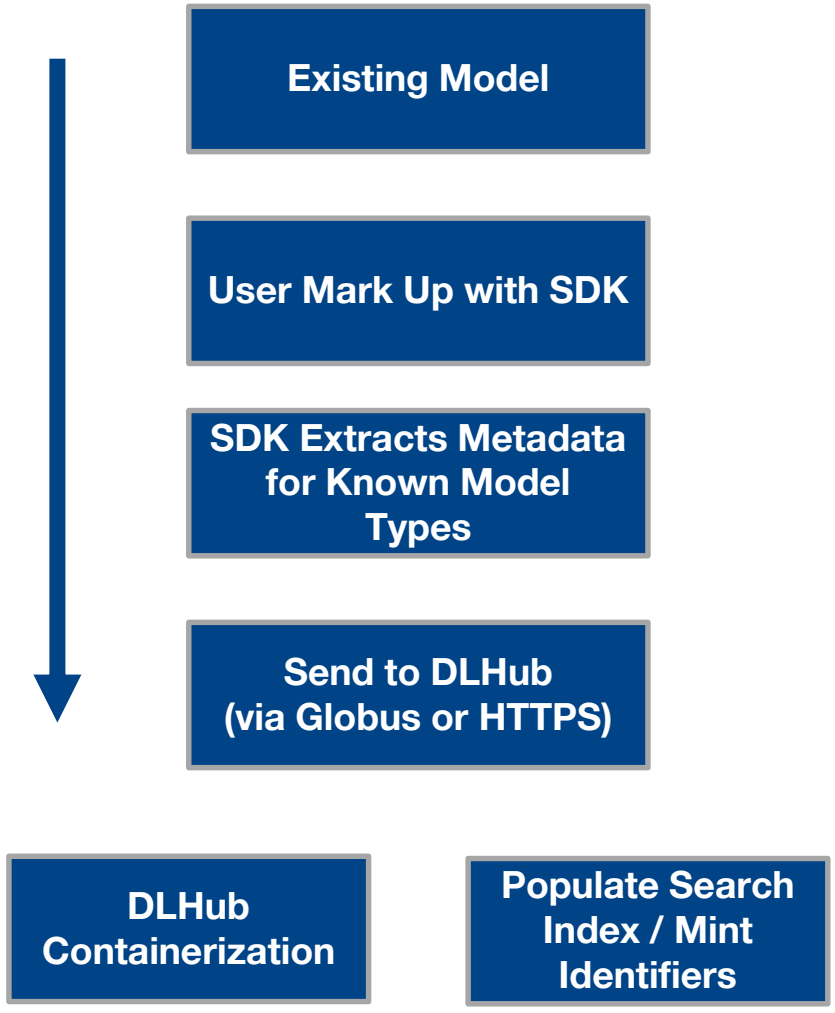
- (future) **Automate** retraining and testing when new data are available



Funding: 2018 Argonne Adv. Computing LDRD

Argonne NATIONAL LABORATORY

Marking up a Model - Python SDK




```python
from dlhub_toolbox.models.servables.keras import KerasModel
import pickle as pkl
import json

# Describe the keras model
model = KerasModel('model.hd5', list(map(str, range(10))))

# Describe the model
model.set_title("MNIST Digit Classifier")
model.set_name("mnist_tiny_example")
model.set_domains(["general","digit recognition"])

# Add link to paper describing the dataset
model.add_related_identifier("10.1109/CVPR.2007.383157", "DOI",
                             "IsDescribedBy")

model.set_authors(["Lecunn, Yann", "Cortes, Corinna"])

# Describe the outputs in more detail
model.output['description'] = 'Probabilities of being 0-9'
model.input['description'] = 'Image of a digit'
```

Python SDK - Automated Metadata Generation

Citation Metadata

```
"datacite": {
    "creators": [{
            "givenName": "Yann",
            "familyName": "Lecunn",
            "affiliations": []
        },
        {
            "givenName": "Corinna",
            "familyName": "Cortes",
            "affiliations": []
        }
    ],
    "titles": [{
        "title": "MNIST Digit Classifier"
    }],
    "publisher": "DLHub",
    "publicationYear": "2018",
    "relatedIdentifiers": [{
        "relatedIdentifier": "10.1109/CVPR.2007.383157",
        "relatedIdentifierType": "DOI",
        "relationType": "IsDescribedBy"
    }],
    "identifier": {
        "identifier": "10.YET/UNASSIGNED",
        "identifierType": "DOI"
    },
    "resourceType": {
        "resourceTypeGeneral": "InteractiveResource"
    }
},
```

DLHub Metadata

```
"dlhub": {
    "version": "0.1",
    "domains": [
        "general",
        "digit recognition"
    ],
    "visible_to": [
        "public"
    ],
    "id": null,
    "name": "mnist_tiny_example",
    "files": {
        "other": [],
        "model": "model.hd5"
    }
},
```

Access Control

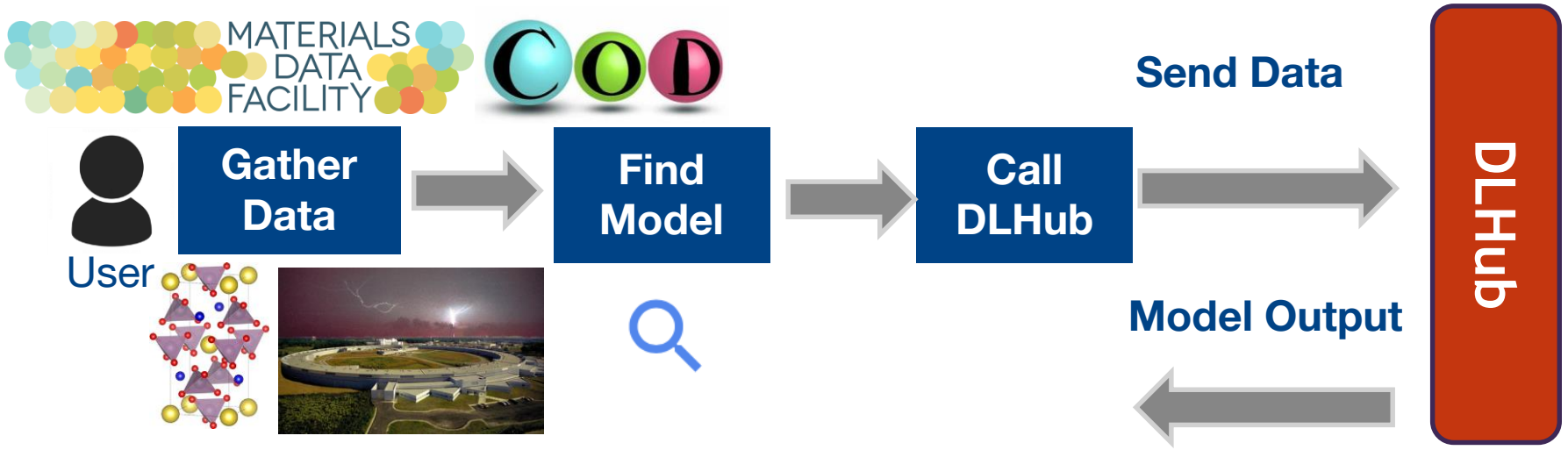
- Public
- Globus users
- Globus groups

Servable Metadata

```
"servable": {
    "methods": {
        "run": {
            "input": {
                "type": "ndarray",
                "description": "Image of a digit",
                "shape": [null,28,28,1]
            },
            "output": {
                "type": "ndarray",
                "description": "Probabilities of being 0-9",
                "shape": [null,10]
            },
            "parameters": {},
            "method_details": {
                "method_name": "predict",
                "classes": ["0","1","2","3","4",
                    "5","6","7","8","9"]
            }
        }
    },
    "shim": "keras.KerasServable",
    "language": "python",
    "dependencies": {
        "python": {
            "keras": "2.2.4",
            "h5py": "2.8.0"
        }
    },
    "type": "Keras Model",
    "model_type": "Deep NN"
```

DLHub Model Discovery and Usage

- Find curated and tested models
- Use models through simple interfaces



Funding: 2018 Argonne Adv. Computing LDRD

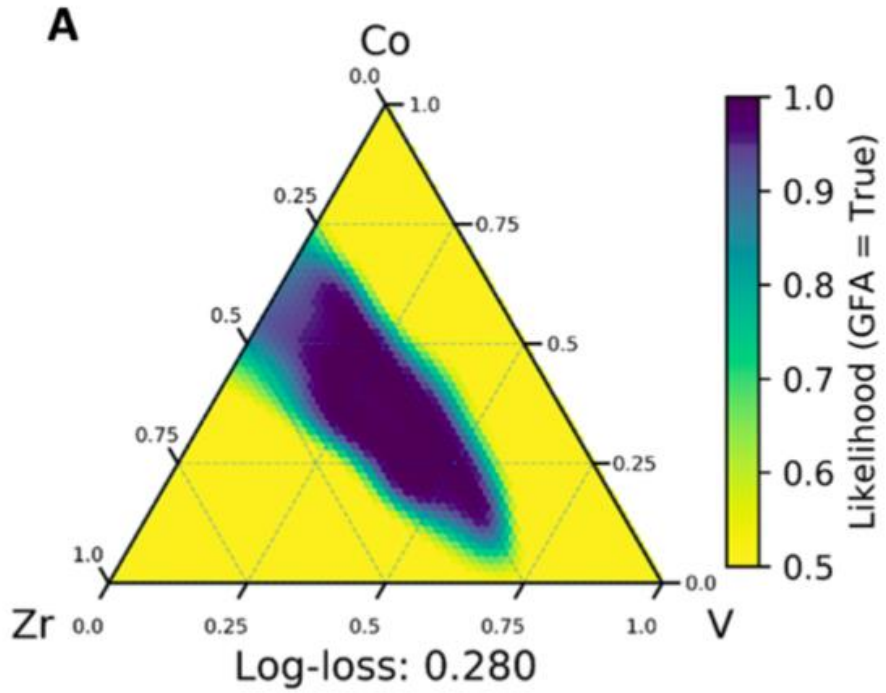
Argonne NATIONAL LABORATORY

Predicting Glass-forming Ability

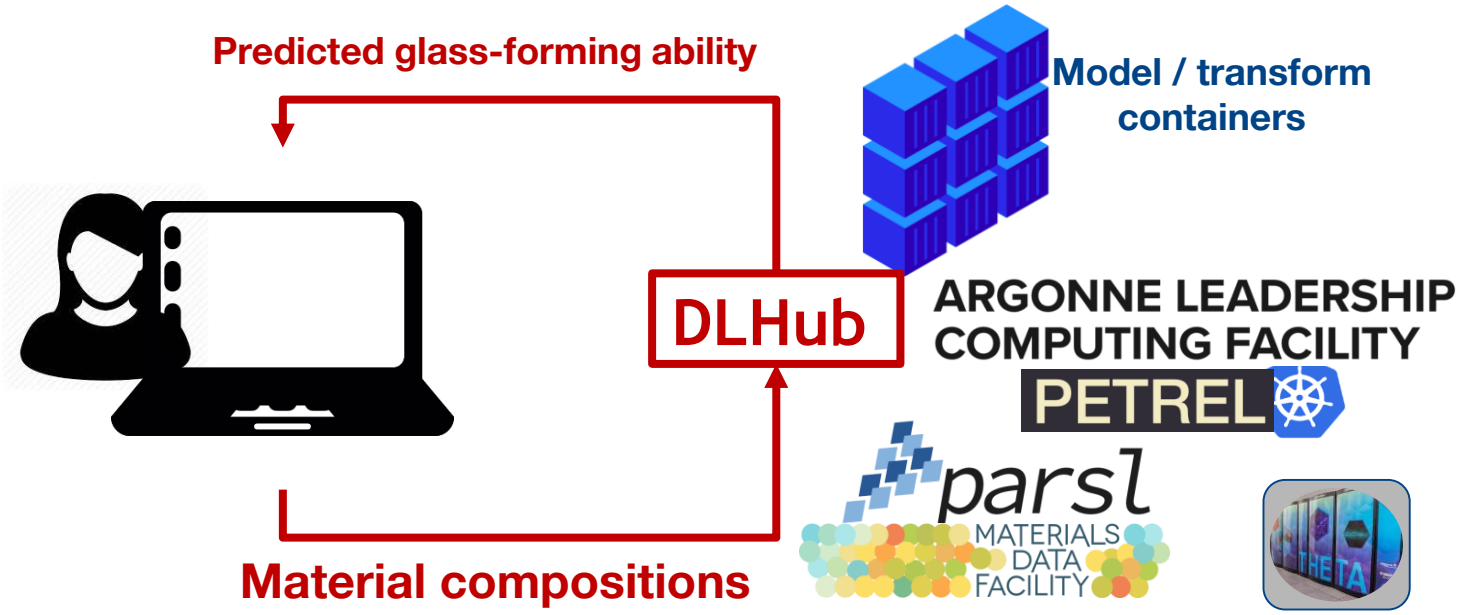
Accelerated discovery of metallic glasses through iteration of machine learning and high-throughput experiments

Fang Ren[1,*], Logan Ward[2,3,*], Travis Williams[4], Kevin J. Laws[5], Christopher Wolverton[2], Jason Hattrick-Simpers[6] and Apurva Mehta[1,†]

10.1126/sciadv.aaq1566



- Where are the model and trained weights?
- How do I run the model on my data?
- How can I retrain the model on new data?
- How can I build on this work?



Funding: 2018 Argonne Adv. Computing LDRD

Argonne NATIONAL LABORATORY

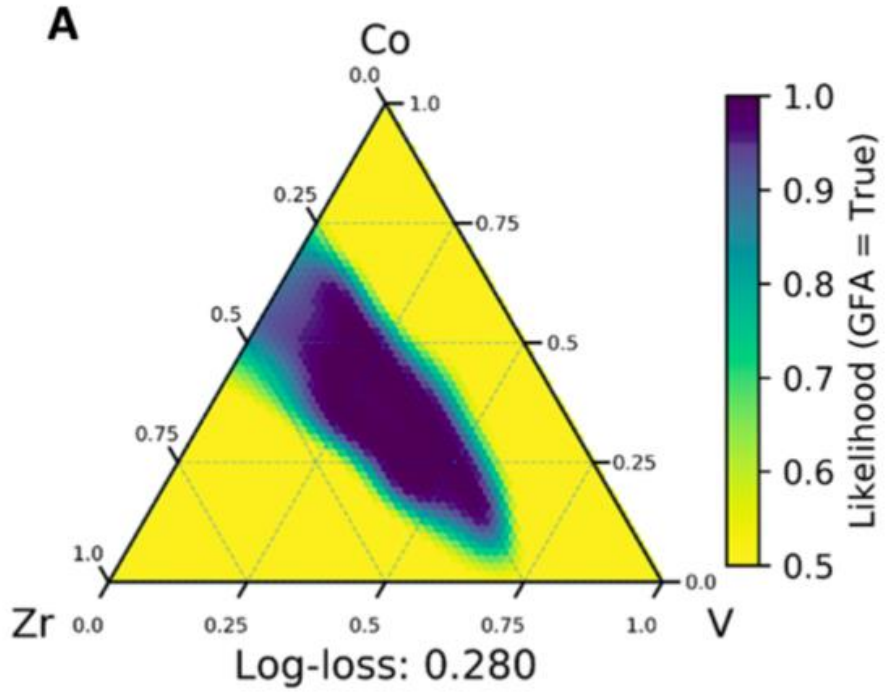
ARGONNE LEADERSHIP COMPUTING FACILITY PETREL parsl

Predicting Glass-forming Ability

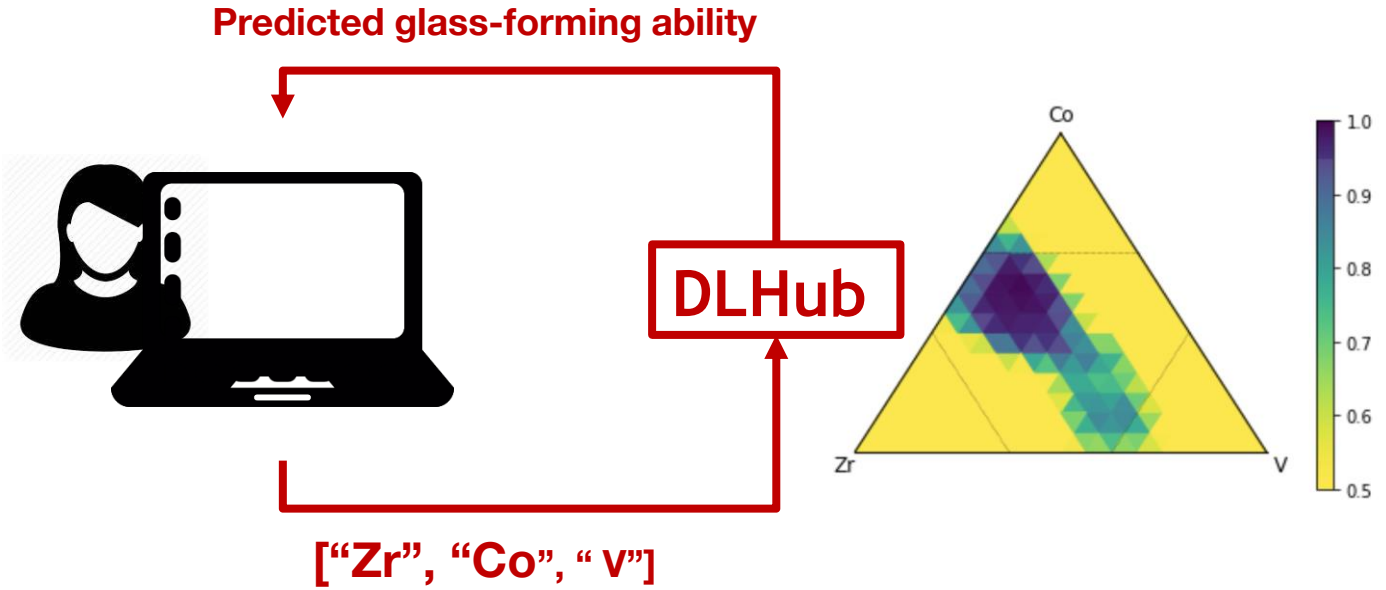
Accelerated discovery of metallic glasses through iteration of machine learning and high-throughput experiments

Fang Ren[1,*], Logan Ward[2,3,*], Travis Williams[4], Kevin J. Laws[5], Christopher Wolverton[2], Jason Hattrick-Simpers[6] and Apurva Mehta[1,†]

10.1126/sciadv.aaq1566




```
servable_name = "metallic_glass"
servable_id = dl.get_id_by_name(servable_name)
elems = ["V","Co","Zr"]

res = dl.run(servable_id, {"data":elems})
```



Funding: 2018 Argonne Adv. Computing LDRD

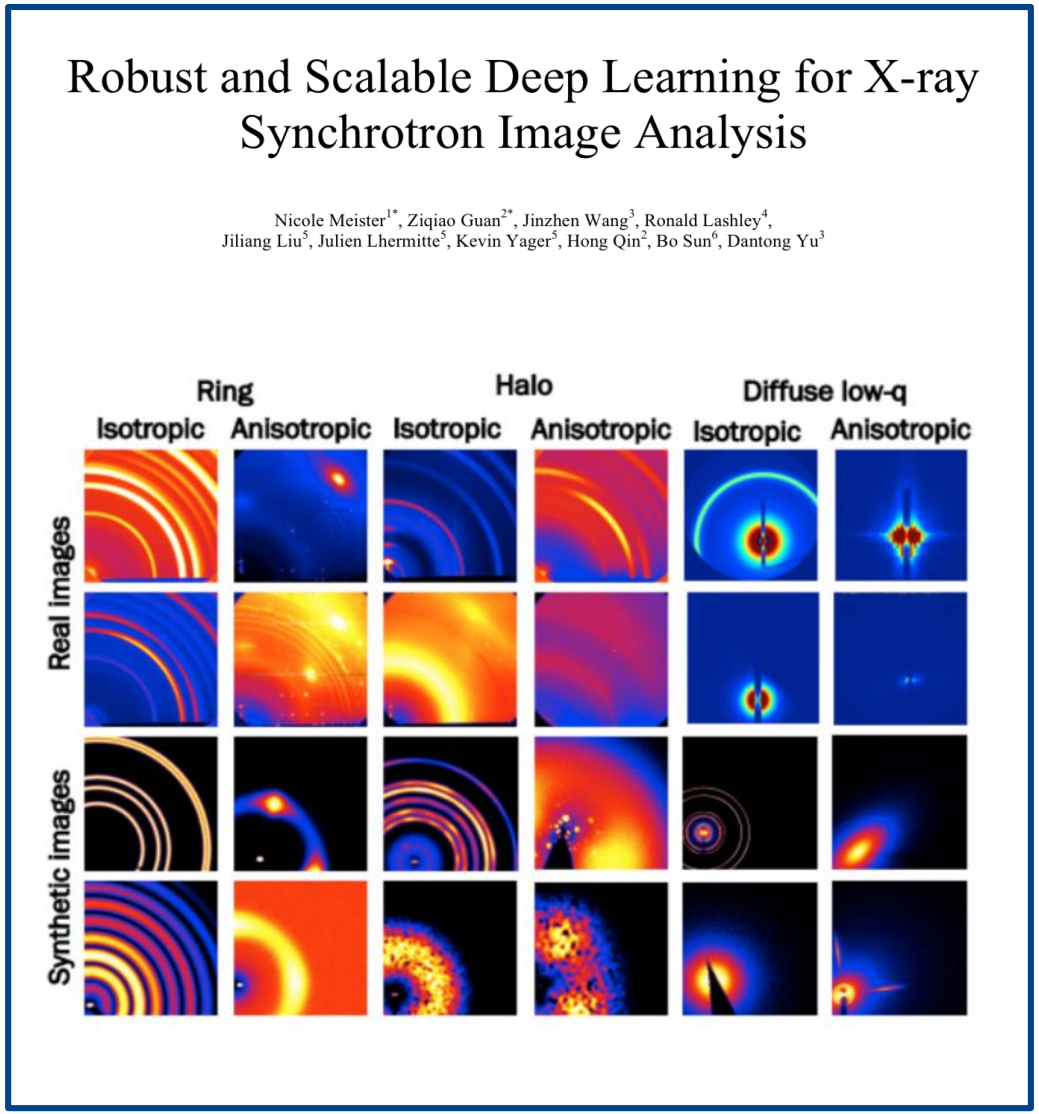
Argonne NATIONAL LABORATORY

ARGONNE LEADERSHIP COMPUTING FACILITY

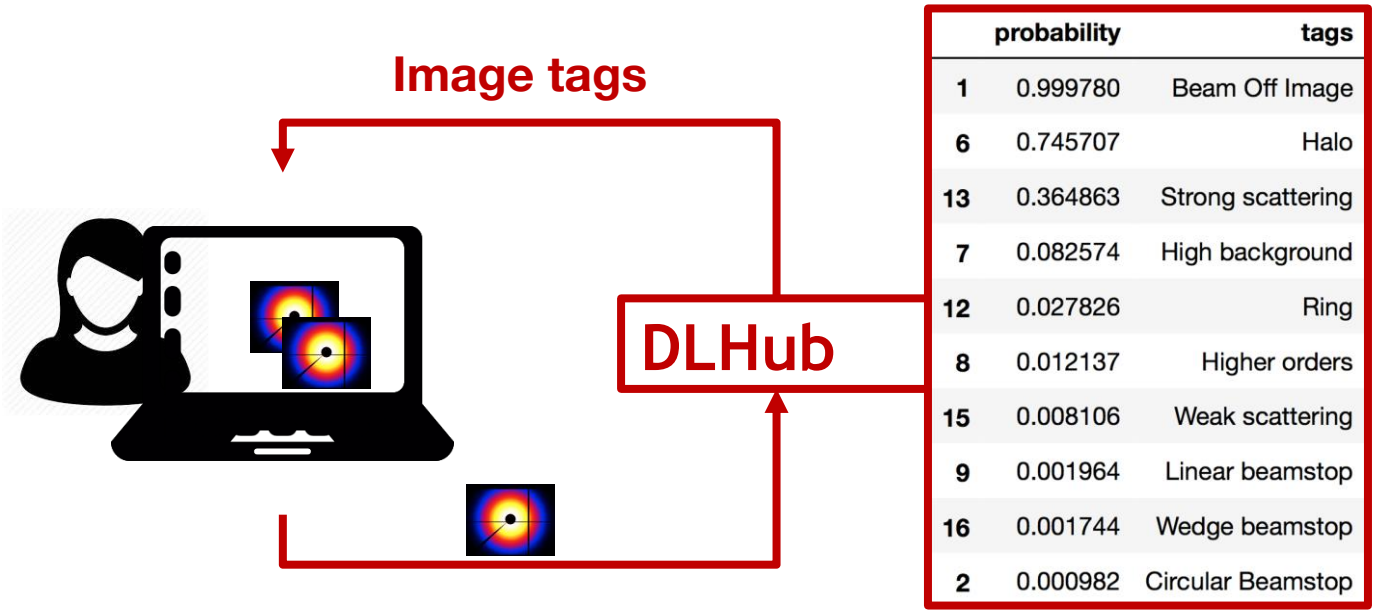
PETREL

parsl

Analyzing Beamline Images



- Stage data into containers via Globus HTTPS
- Pass valid token and data location



	probability	tags
1	0.999780	Beam Off Image
6	0.745707	Halo
13	0.364863	Strong scattering
7	0.082574	High background
12	0.027826	Ring
8	0.012137	Higher orders
15	0.008106	Weak scattering
9	0.001964	Linear beamstop
16	0.001744	Wedge beamstop
2	0.000982	Circular Beamstop

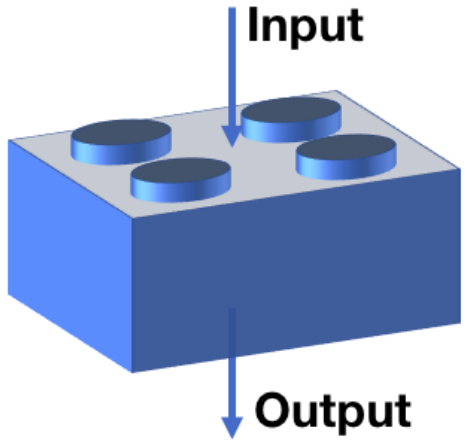
Funding: 2018 Argonne Adv. Computing LDRD

Argonne NATIONAL LABORATORY

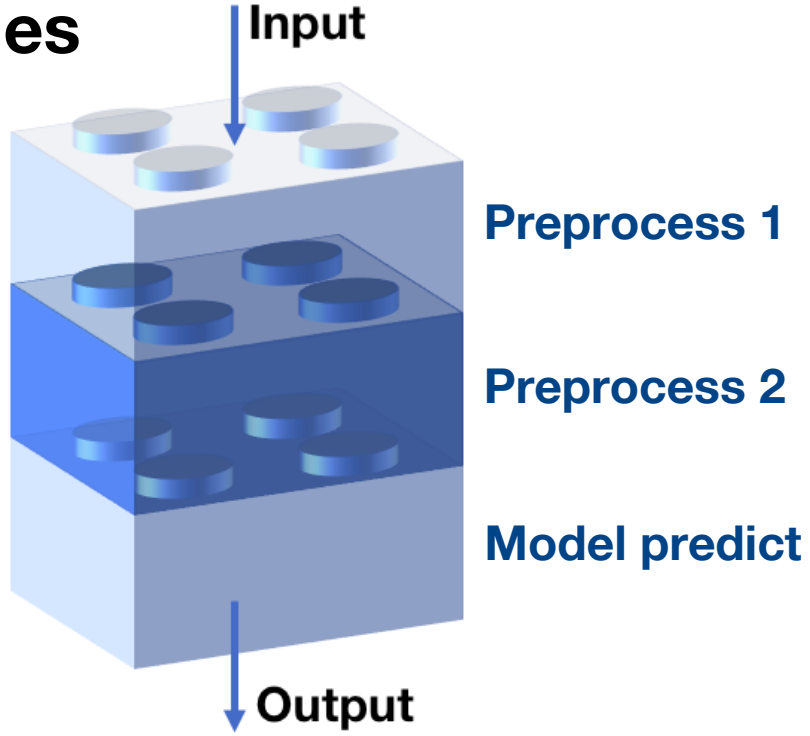
ARGONNE LEADERSHIP COMPUTING FACILITY PETREL parsl

Data and Learning Hub (DLHub): Pipelines

Servables



Pipelines



Funding: 2018 Argonne Adv. Computing LDRD

Argonne NATIONAL LABORATORY

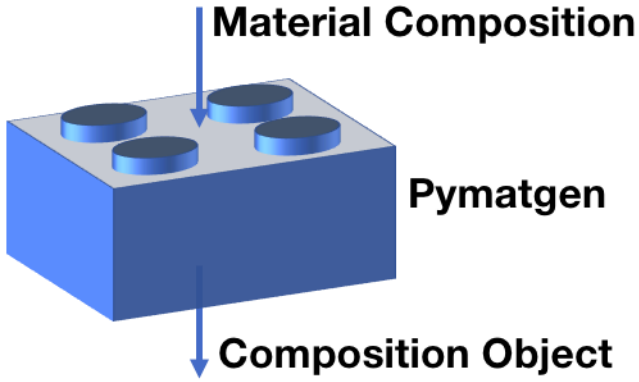
ARGONNE LEADERSHIP COMPUTING FACILITY

PETREL

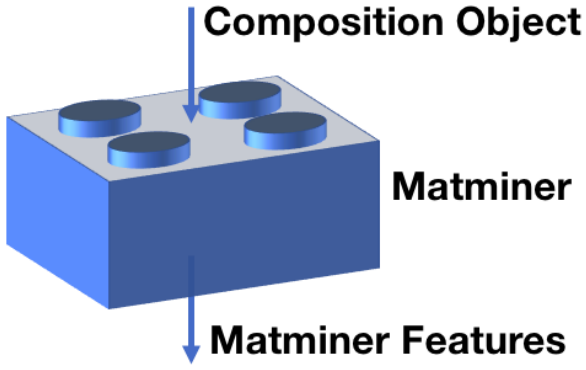
parsl

Pipelines: Predicting Formation Enthalpy

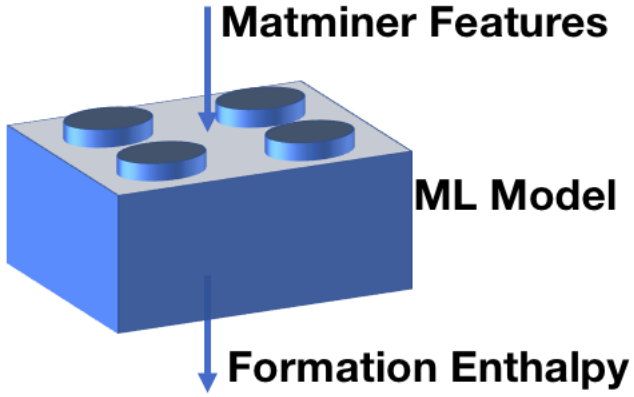
Step 1



Step 2



Step 3

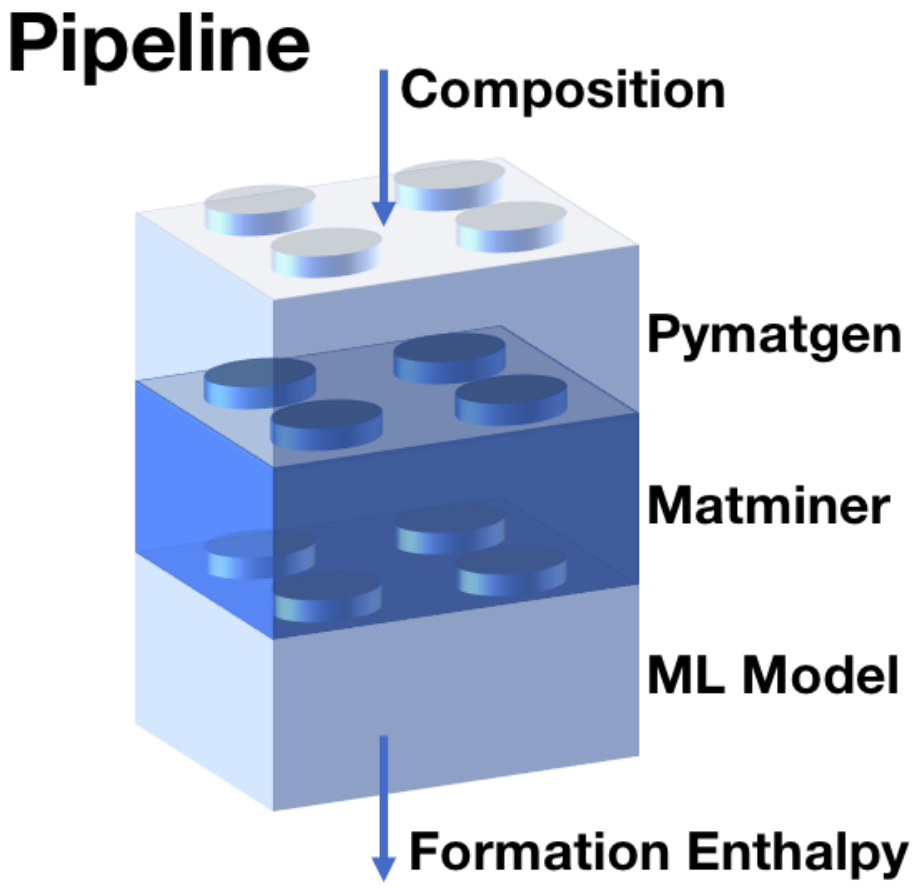


Funding: 2018 Argonne Adv. Computing LDRD

Argonne NATIONAL LABORATORY

ARGONNE LEADERSHIP COMPUTING FACILITY PETREL parsl

Predicting Formation Enthalpy



Funding: 2018 Argonne Adv. Computing LDRD

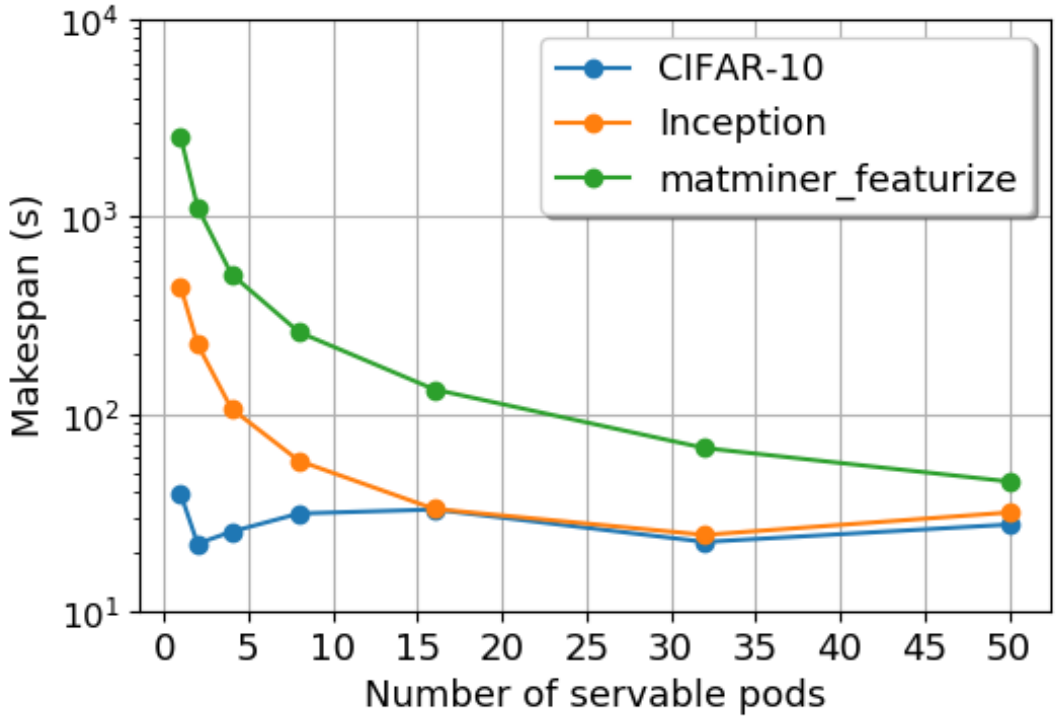
Argonne NATIONAL LABORATORY

ARGONNE LEADERSHIP COMPUTING FACILITY PETREL parsl

DLHub Performance

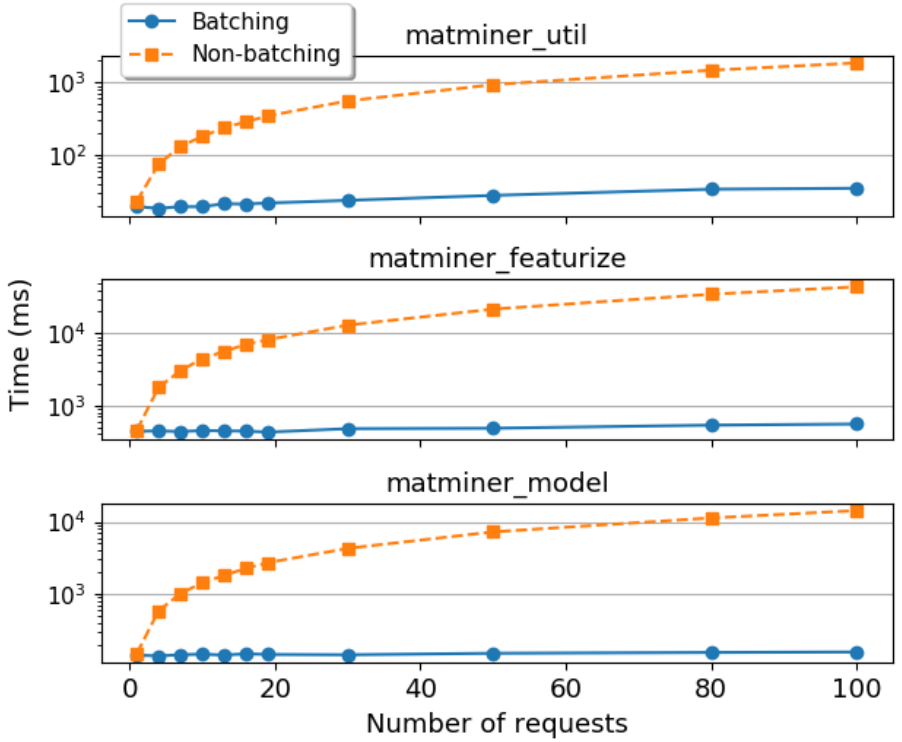
DLHub Performance

Scale Testing



The time required for the Inception, CIFAR10, and Matminer-featurize models to process 5000 inferences with varying numbers of replicas.

Batching



Servable invocation time, with and without batching.

Funding: 2018 Argonne Adv. Computing LDRD

Argonne NATIONAL LABORATORY

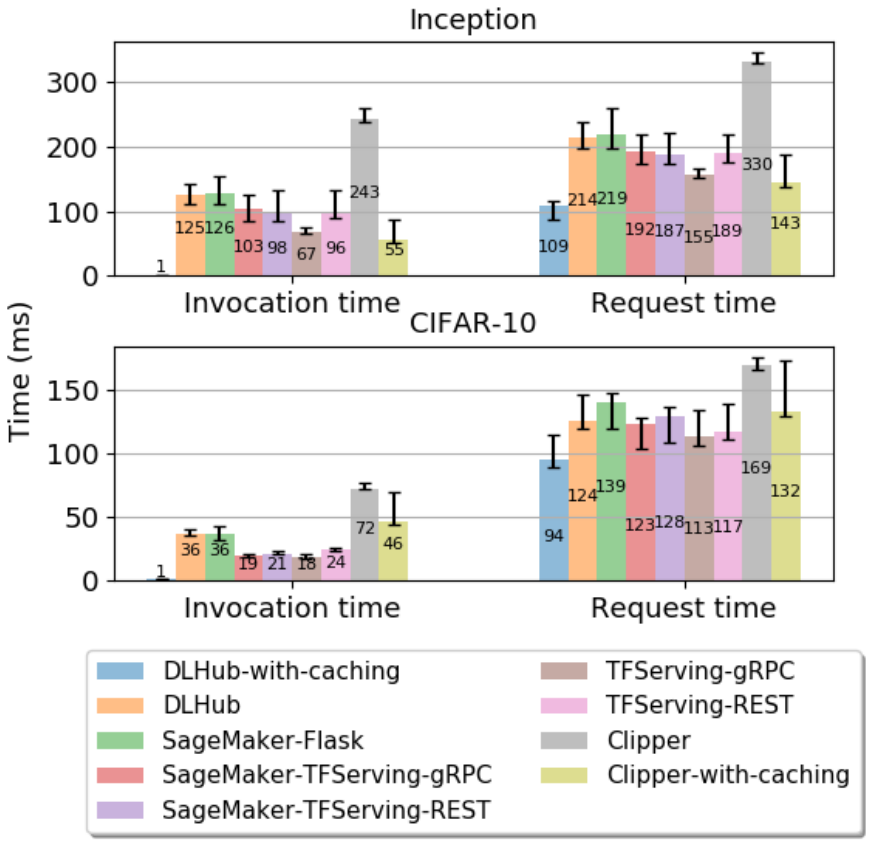
ARGONNE LEADERSHIP COMPUTING FACILITY

PETREL

parsl

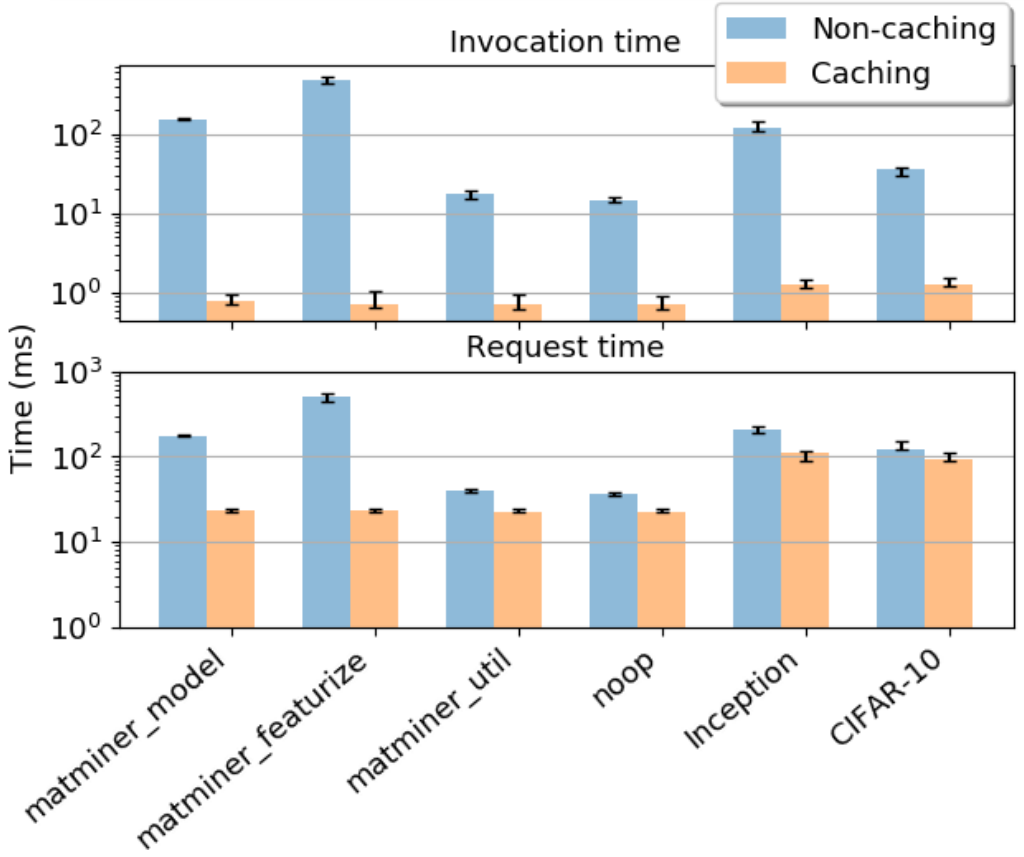
DLHub Performance

Serving General Models



Performance of different serving systems on the Inception and CIFAR-10 problems.

Caching



Performance impact of caching in DLHub. Bars and error bars show median and 5th/95th percentiles

Funding: 2018 Argonne Adv. Computing LDRD

Argonne NATIONAL LABORATORY

ARGONNE LEADERSHIP COMPUTING FACILITY

PETREL

parsl

DLHub Summary

Model deposit and discovery

- Developed a model schema to promote discovery
- Implemented advanced search and filtering
- Built ingest flow: models are dynamically staged, packaged, dockerized, published, and indexed

Model serving

- Deployed capabilities for users to run inference with SDK and CLI
- Automated testing of containers
- Implemented caching and batching

Support for multiple execution sites

- PetrelKube: Parsl, TF serving, Sagemaker
- Other: AWS, OSG

Authentication

- Protected model metadta and inference with GlobusAuth
- Secured data staging

Monitoring and statistics

- Request, invocation, data staging

Future work

- Dynamic scaling by load
- Build Web UI to create pipelines and invoke models
- Cache at the servable level within pipelines
- Couple DLHub to data sources (MDF, etc.)
- Integrate with ML frontend tools (DeepForge), optimization tools (DeepHyper), and more
- Create interface for training and retraining of models

Funding: 2018 Argonne Adv. Computing LDRD

Argonne NATIONAL LABORATORY

ARGONNE LEADERSHIP COMPUTING FACILITY PETREL parsl

Thanks to our sponsors!

ARGONNE LEADERSHIP COMPUTING FACILITY

ALCF DF

PETREL

Data Management and Sharing Pilot

Parsl

NSF

Globus

NIST

U.S. DEPARTMENT OF ENERGY

NATIONAL INSTITUTES OF HEALTH

NSF

MATERIALS DATA FACILITY

IMaD

NSF

NIST

CHiMaD

Center for Hierarchical Materials Design

DLHub

U.S. DEPARTMENT OF ENERGY

Argonne LDRD

U.S. DEPARTMENT OF ENERGY

Argonne NATIONAL LABORATORY

THE UNIVERSITY OF CHICAGO