NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

Artificial Intelligence for Materials Science Workshop: Hands-On Session

Daniel Wines, Brian DeCost, Kamal Choudhary

AIMS: July 10, 2025

https://jarvis.nist.gov

https://github.com/usnistgov/aims_workshop/

Joint Automated Repository for Various Integrated Simulations

7/10/2025

Outline

https://github.com/usnistgov/aims_workshop

JARVIS-DFT

NIST MATERIAL MEASUREMENT LABORATORY

jarvis.nist.gov/jarvisdft/

NIST

JARVIS Publications Colab Downloads Tutorials Events Log In / Sign Up Help

Density Functional Theory (DFT)

Click on periodic table elements (e.g., Ni-Al-) or enter a chemical formula (e.g., Al2O3) or enter a JARVIS-ID (e.g., JVASP-1002) and click Search.

Mo-S- Search Refresh

https://jarvis.nist.gov/jarvisdft/

how 100 entries Search:

id	formula	Spg	SpgNum	crys	func	E_form	OPT_gap	MBJ_gap	hse_gap	Kv	Gv	poisson	spillage	slme	mag	type
JVASP-664	MoS2	P-6m2	187	hexagonal	OptB88vdW	-0.88454	1.658	-	-	-	-	-	0.003	-	0.0	2D
JVASP-730	MoS2	R-3m	166	trigonal	OptB88vdW	-0.60967	0.000	-	-	-	-	-	0.976	-	-0.0	2D
JVASP-100856	MoS3	P2_1/m	11	monoclinic	OptB88vdW	-0.34773	0.443	-	-	-	-	-	-	-	0.0	3D
JVASP-111117	MoS2	P-3m1	164	trigonal	OptB88vdW	-0.69305	0.000	0.0	-	-	-	-	-	-	0.0	3D
JVASP-127507	MoS	Pm-3m	221	cubic	OptB88vdW	-0.18293	0.000	-	-	-	-	-	-	-	0.0	3D
JVASP-132473	MoS3I	P1	1	triclinic	OptB88vdW	0.06061	0.651	-	-	-	-	-	-	-	3.993	3D
JVASP-144515	Mo3S4	P-1	2	triclinic	OptB88vdW	-0.73826	0.000	-	-	-	-	-	-	-	0.0	3D
JVASP-228	MoS2	R-3m	166	trigonal	OptB88vdW	-0.69909	0.000	0.0	-	51.32	45.47	0.19	0.852	-	0.0	3D
JVASP-28379	MoS2	P6_3/mmc	194	hexagonal	OptB88vdW	-0.95754	0.958	1.364	-	71.48	44.27	0.26	-	33.92	-	3D
JVASP-28413	MoS2	P-3m1	164	trigonal	OptB88vdW	-0.92268	1.207	1.471	-	-	-	-	-	32.7	0.0	3D
JVASP-28733	MoS2	P6_3/mmc	194	hexagonal	OptB88vdW	-0.96160	0.921	-	-	70.62	47.69	0.23	0.013	-	0.0	3D
JVASP-34138	MoS2	P6_3/mmc	194	hexagonal	OptB88vdW	-0.96135	0.920	1.356	-	-	-	-	-	33.93	0.0	3D

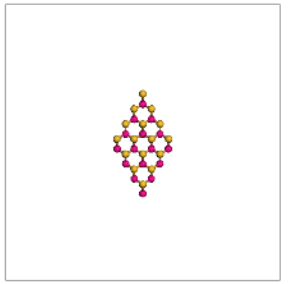
JARVIS API JARVIS-DFT JARVIS-ML JARVIS-FF JARVIS-Tools Documentation Publications Report bug/Contact

Structure XRD DOS Bands Spillage Optics(GGA) Elastic Thermoelectric Convergence

ID: JVASP-664	Functional: OptB88vdW	Primitive cell	Primitive cell	Conventional cell	Conventional cell
Chemical formula: MoS2	Formation energy/atom (eV): -0.88454	a 3.19 Å	α: 90.0 °	a 3.19 Å	α: 90.0 °
Space-group:P-6m2 (187)	Relaxed energy/atom (eV):-5.21029	b 3.19 Å	β: 90.0 °	b 3.19 Å	β: 90.0 °
Crystal system:hexagonal	Point group:-6m2	c 34.88 Å	γ: 120.0 °	c 34.88 Å	γ: 120.0 °
Data source:JARVIS-DFT-VASP	Material type:SingleLayer	Density (gcm⁻³):0.866	Volume (Å³):306.988	nAtoms_prim:3	nAtoms_conv:3
SCF direct bandgap (eV):1.683	SCF indirect bandgap (eV):1.658	Magnetic moment (μB):0.0	Exfoliation energy (meV/atom):	Packing fraction:0.069	Number of species:2
Band direct gap (eV):1.71	Band indirect gap (eV):1.71	TBmBJ direct gap (eV):	TBmBJ indirect gap (eV):	HSE06 direct gap (eV):2.49	HSE06 indirect gap (eV):2.36
Voigt bulk mod. (GPa):	Voigt shear mod. (GPa):	Poisson ratio:	Anisotropy ratio:	Solar SLME (%):	Solar SQ (%):
Max. IR mode (cm⁻¹):	Max. Raman mode (cm⁻¹):	Min. IR mode (cm⁻¹):	Min. FD phonon (cm⁻¹):-0.0	Cut-off (eV):650	K-point length (Å):25

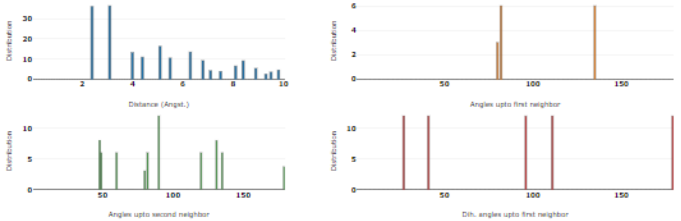
Show POSCAR Show POSCAR-conv Show XYZ format Show CIF format

Visualizing Atomic structure



Atomic Structure Analysis

The following shows the radial, angle and dihedral distribution function plots.



JARVIS-DFT

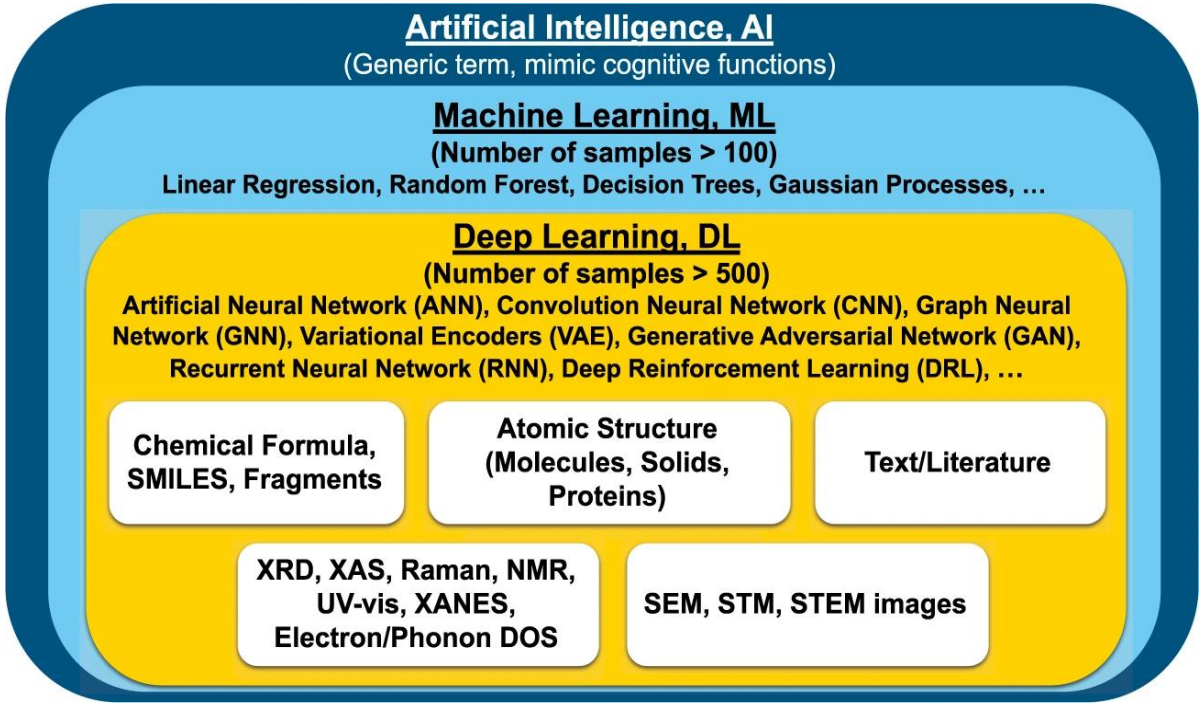
Features	JARVIS-DFT
#Materials (Struct., E_f, E_g)	80000
DFT functional/methods	vdW-DFT-OptB88, TBmBJ, DFT+SOC
K-point/cut-off	Converged for each material
SCF convergence criteria	Energy & Forces
Elastic tensors & point phonos	17402
Piezoelectric, IR spect.	4801
Dielectric tensors (w/o ion)	4801 (15860)
Electric field gradients	11865
SuperCon Tc	2200 (1058 ambient condition)

Features	JARVIS-DFT
2D monolayers	1011
Raman spectra	400
Seebeck, Power Factors	23210
Solar SLME	8614
Spin-orbit Coupling Spillage	11383
WannierTB	1771
STM images	1432
Surfaces	300
Defects	400
Interfaces	1.4 trillion (IU), 600 (ASJ)

Hands-on: JARVIS-DFT Analysis

1 https://colab.research.google.com/github/knc6/jarvis-tools-notebooks/blob/master/jarvis-tools-notebooks/Analyzing_data_in_the_JARVIS_DFT_dataset.ipynb
2 https://colab.research.google.com/github/knc6/jarvis-tools-notebooks/blob/master/jarvis-tools-notebooks/Basic_ML.ipynb
3 https://colab.research.google.com/github/knc6/jarvis-tools-notebooks/blob/master/jarvis-tools-notebooks/alignn_jarvis_leaderboard.ipynb
4 Brian Notebook on MLFF
5 CHIPS-FF Notebook
6 https://colab.research.google.com/github/knc6/jarvis-tools-notebooks/blob/master/jarvis-tools-notebooks/atomgpt_example.ipynb
https://colab.research.google.com/github/knc6/jarvis-tools-notebooks/blob/master/jarvis-tools-notebooks/DiffractGPT_example.ipynb
https://colab.research.google.com/github/knc6/jarvis-tools-notebooks/blob/master/jarvis-tools-notebooks/MicroscopyGPT.ipynb

Deep learning for materials



npj | computational materials

Explore content · About the journal · Publish with us

nature › npj computational materials › review articles › article

Review Article | Open Access | Published: 05 April 2022

Recent advances and applications of deep learning methods in materials science

Kamal Choudhary, Brian DeCost, Chi Chen, Anubhav Jain, Francesca Tavazza, Ryan Cohn, Cheol Woo Park, Alok Choudhary, Ankit Agrawal, Simon J. L. Billinge, Elizabeth Holm, Shyue Ping Ong & Chris Wolverton

npj Computational Materials **8**, Article number: 59 (2022) | Cite this article

REVIEW ARTICLE | OCTOBER 18 2023

Recent progress in the JARVIS infrastructure for next-generation data-driven materials design

Daniel Wines ; Ramya Gurunathan ; Kevin F. Garrity ; Brian DeCost ; Adam J. Biacchi ; Francesca Tavazza ; Kamal Choudhary

+ Author & Article Information

Appl. Phys. Rev. 10, 041302 (2023)

https://doi.org/10.1063/5.0159299 **Article history**

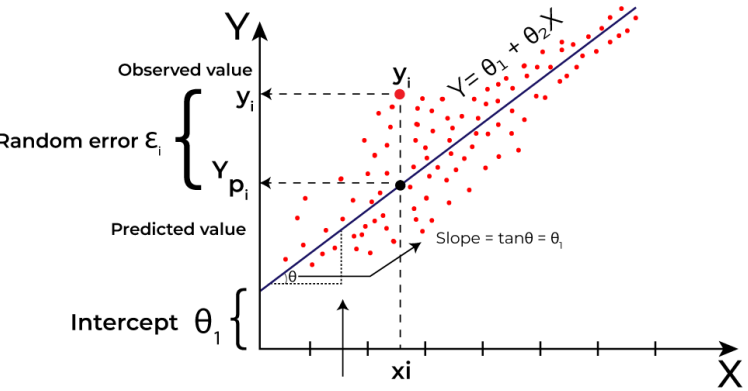
Tour of Machine Learning Models

- **Dimensionality reduction**: feature extraction technique that aims to reduce the number of input features
  - PCA, t-SNE, UMAP
- **Instance methods**: Compares instances in data with a similarity measure to identify best matches
  - K-Nearest Neighbor, Self-Organizing Maps (SOM)
- **Regression**: Establish relationship between independent variable X and dependent variable Y by iteratively optimizing errors made in predictions, most of these can be used for classification as well
  - Logistic Regression, Neural networks, Decision trees
- **Clustering Models**: describes the class of problem, different from classification, works with unlabeled data
  - Hierarchical Clustering, k-Means
- **Regularization models**: penalizes models based on their complexity
  - Ridge regression, LASSO, Elatic Net
- **Bayesian algorithms**: apply Bayes' Theorem for problems
  - Naïve Bayes, Bayesian Network
- **Ensemble algorithms**: models composed of multiple weaker models that are independently trained and whose predictions are combined in some way to make the overall predictions better
  - Random Forest, Gradient Boosted Regression Trees, AdaBoost
- **Neural Network Algorithms**: inspired by the structure and/or function of biological neural networks
  - Multilayer Perceptrons, convolution NN, Graph NN, transformers, Auto-encoders, diffusion models

https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/ ,
https://www.geeksforgeeks.org/machine-learning-algorithms/

ML Models

Linear Regression



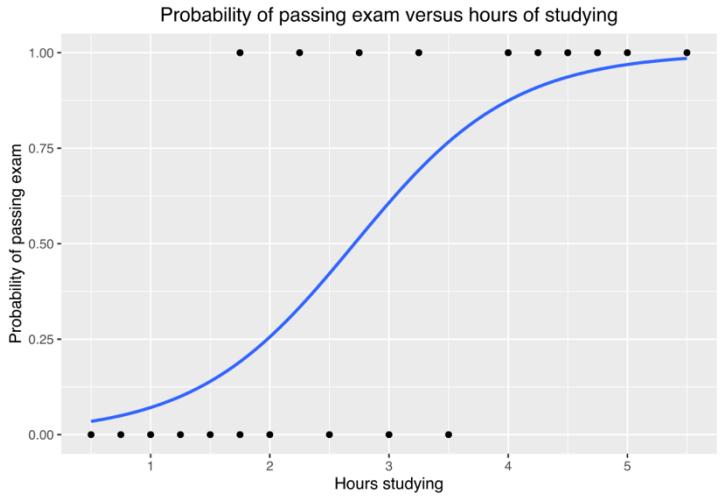
$$y = \beta_0 + \beta_1 x$$

- y is the dependent variable.
- x is the independent variable.
- β0 is the y-intercept of the line.
- β1 is the slope of the line.
- To estimate $\beta 0$ and $\beta 1$, we minimize the sum of the squared residuals:

$$\text{RSS} = \sum_{i=1}^{n}(y_i - (\beta_0 + \beta_1 x_i))^2$$

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

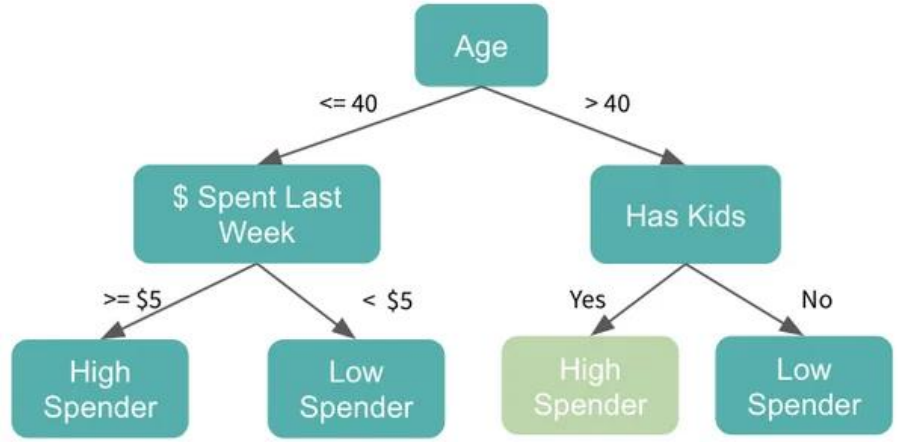
Logistic Regression



- Log-odds of an event as a linear combination of one or more independent variables
- Used for predicting a binary or categorical dependent variable.

$$p(x) = \frac{1}{1 + e^{-(x-\mu)/s}}$$

Tree based methods



Tree-based models use a series of if-then rules to generate predictions from one or more decision trees: RandomForest, GBM

Gini Impurity:

$$Gini = 1 - \sum_{i=1}^{n} p_i^2$$

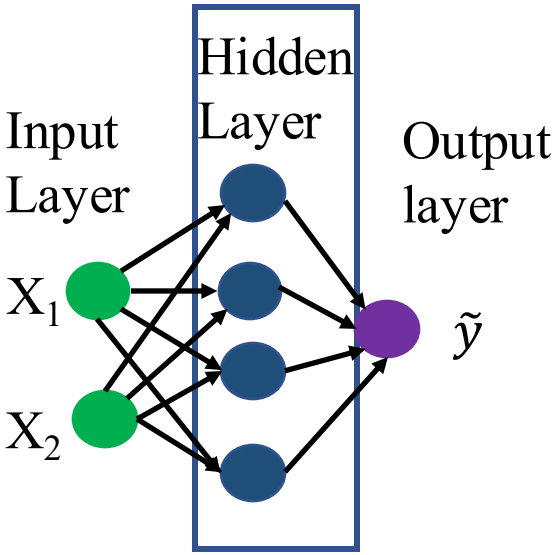
Where p_i is the proportion of samples belonging to class i in the node.

$$Entropy = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

Deep learning for materials

NIST MATERIAL MEASUREMENT LABORATORY

Standard NN



1) Forward propagation

$$z^{[l]} = W^{[l]} a^{[l-1]} + b^{[l]}$$

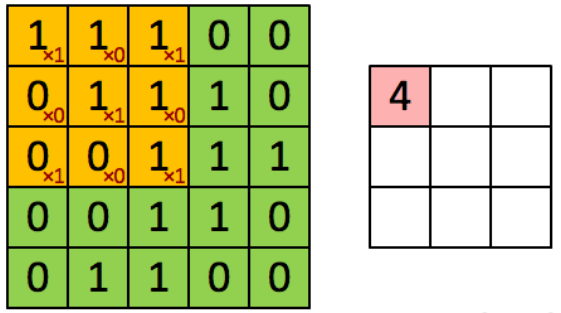
$$a^{[1]} = \sigma(z^{[l]}); \quad a^{[0]} = X$$

2) $Cost, J(W, b) = f(y - \tilde{y})$

3) Gradient descent (∇J):
minimize cost with W,b

4) Backpropagation:
chain rule to get, $\frac{\partial J}{\partial W}$

ConvolutionNN



1	0	1
0	1	0
1	0	1

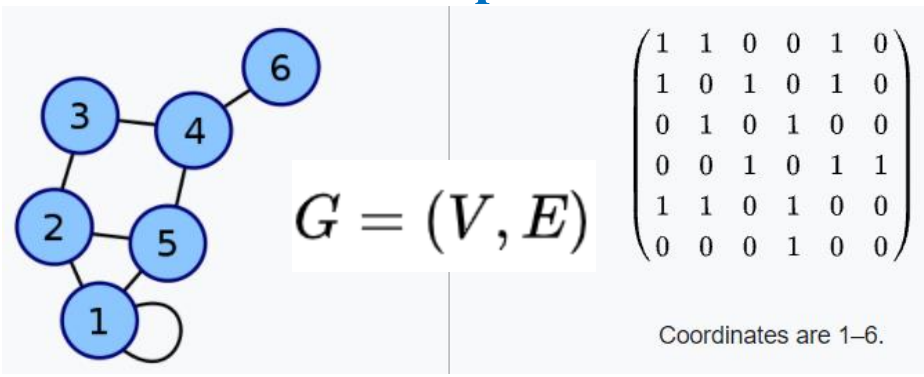
1) Convolution:
element-wise multiplication & sum

2) Pool: Max, Average, Sum

3) Fully Connected: Standard NN

Shared weights (Learnable filters), regularized version of NNs

GraphConvNN



Types: un/weighted, un/directed, line, Hetero/Homogenous, Multigraph

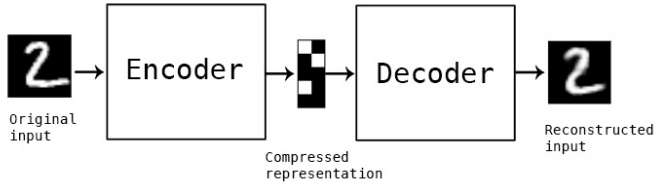
1) Adjacency matrix, N x N (N: #nodes),
2) D: degree of node
3) Update node representation using message passing, GPU efficient
4) Update equation is local, neighborhood of a node only, independent of graph size

$$h_i^{\ell+1} = f(h_i^{\ell}, \{h_j^{\ell}\}_{j \in \mathcal{N}_i})$$

Deep learning for materials

NIST MATERIAL MEASUREMENT LABORATORY

AutoEncoders



Used for Dimensionality reduction or feature learning:

1) Encoding functions: *x* to a latent dimension *z*

$$\mathbf{z} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$$

2) Decoding functions: *z* to x

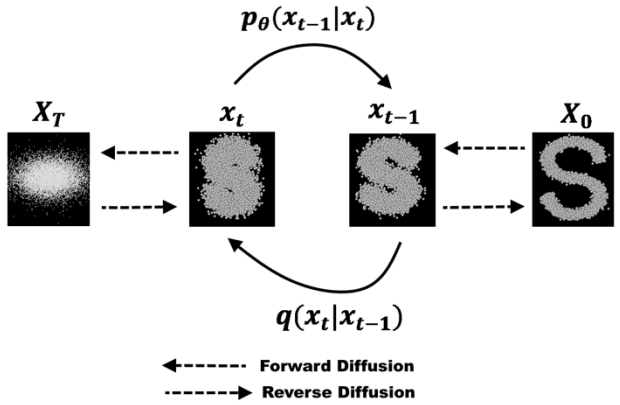
$$\mathbf{x}' = \sigma'(\mathbf{W}'\mathbf{z} + \mathbf{b}')$$

3) Loss function

$$L(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|^2$$

$$\|\mathbf{x} - \sigma'(\mathbf{W}'(\sigma(\mathbf{W}\mathbf{x} + \mathbf{b})) + \mathbf{b}')\|^2$$

Diffusion Models



Reverse diffusion process, starting from random noise

1)Forward diffusion: gradually adds noise over time steps

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \prod_{t=1}^{T} \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

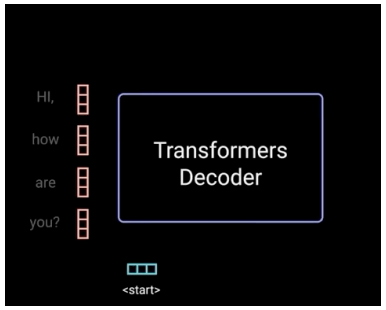
2) Reverse Diffusion Process:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

3) Training Objective and sampling:

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}[\, \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\|^2 \,]$$

Transformers



1) Self-attention: calculates the relevance between each pair of input tokens.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

2) Multi-head attention: focus on different parts of the input sequence simultaneously

3) $$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O$$

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

4) Positional Encoding: instead of RNN, use PE to convey the order of elements in the sequence

$$\text{PE}_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right)$$

$$\text{PE}_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right)$$

ALIGNN

NIST MATERIAL MEASUREMENT LABORATORY

Atomistic Line Graph Neural Network

github.com/usnistgov/alignn

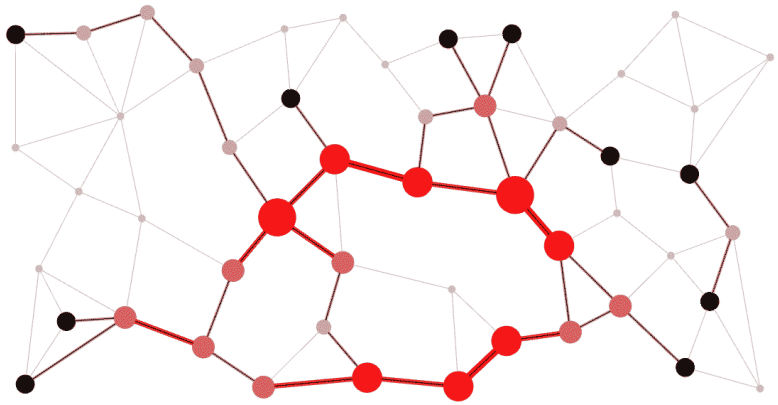
usnistgov / alignn

Type / to

Code | Issues 23 | Pull requests 5 | Discussions | Actions | Proje

alignn Public

Edit Pins



npj | computational materials

Explore content | About the journal | Publish with us

nature > npj computational materials > articles > article

Article | Open Access | Published: 15 November 2021

Atomistic Line Graph Neural Network for improved materials property predictions

Kamal Choudhary & Brian DeCost

npj Computational Materials 7, Article number: 185 (2021) | Cite this article

Digital Discovery

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal

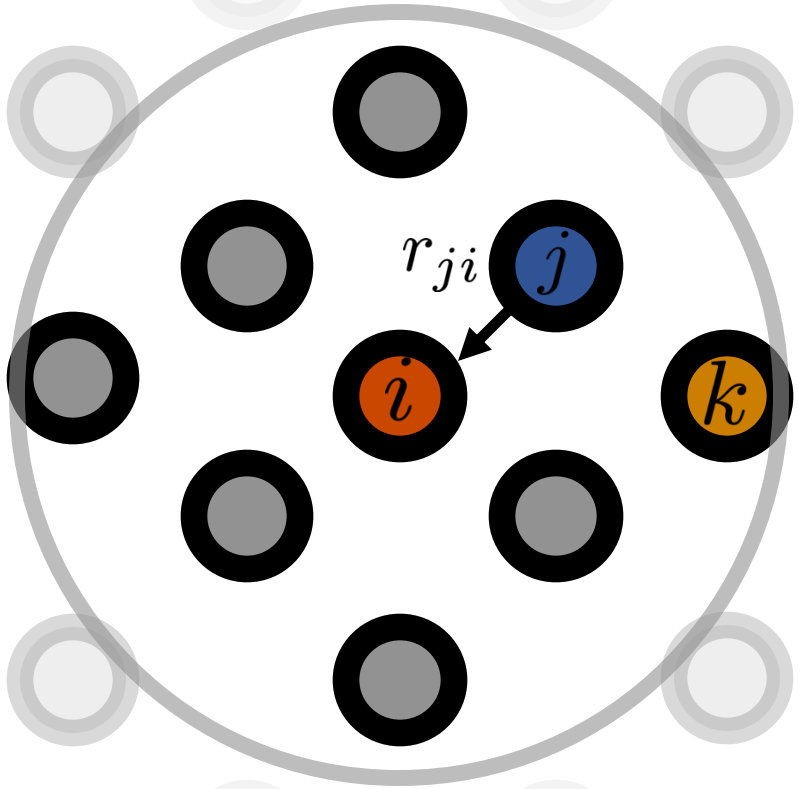
Check for updates

Cite this: DOI: 10.1039/d2dd00096b

Unified graph neural network force-field for the periodic table: solid state applications

Kamal Choudhary, *ab Brian DeCost, c Lily Major, de Keith Butler, e Jeyan Thiyagalingam e and Francesca Tavazza c

Interatomic potentials



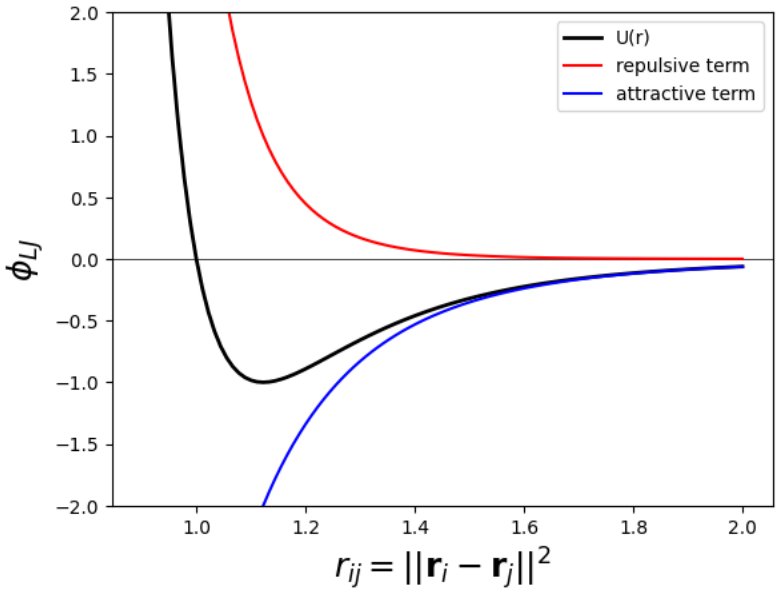
Expand the potential U({r}) in terms of n-body interactions

$$U(\{\mathbf{r}\}) = \sum_i \phi_1(\mathbf{r}_i) + \sum_i \sum_j \phi_2(\mathbf{r}_i, \mathbf{r}_j) + \sum_i \sum_j \sum_k \phi_3(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) + \dots$$

Lennard Jones potential: repulsive and attractive terms

$$U(\{\mathbf{r}\}) = \sum_i \sum_j \phi_{LJ}(r_{ij})$$

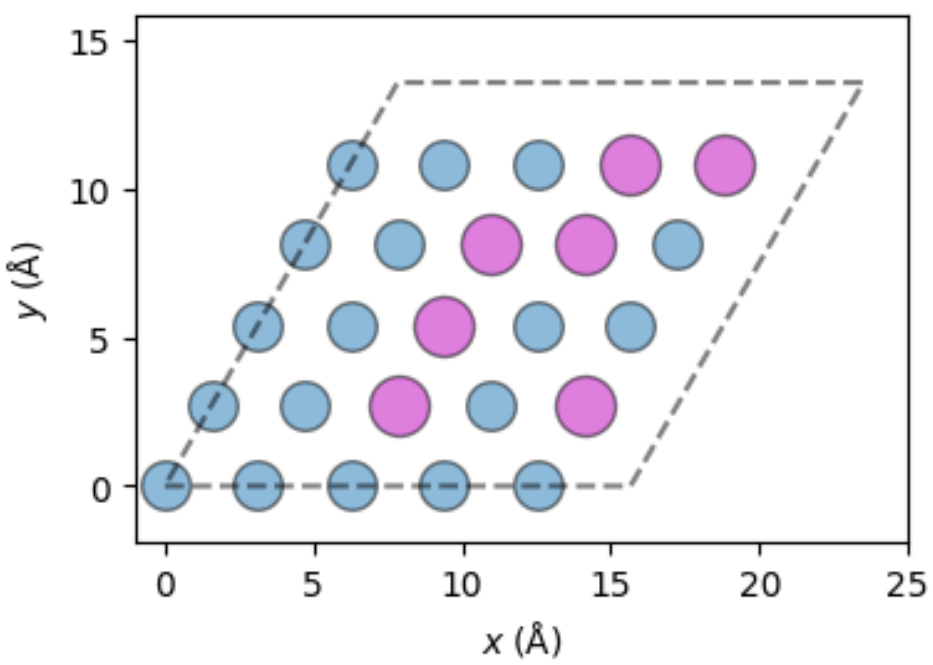
$$\phi_{LJ}(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right]$$

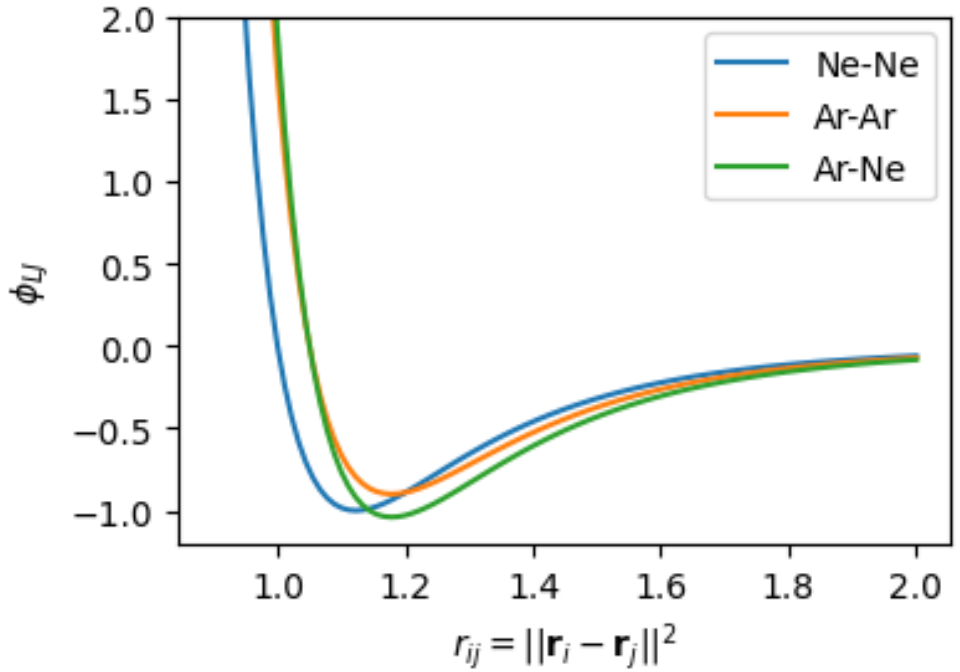


Multicomponent systems

For example, Ne-Ar system

Add separately-parameterized pair potentials for each combination of species





1. Model size scales unfavorably with species considered, especially with more complex potential forms
2. Low model capacity limits ability to capture complex environment-dependent bonding

From pair potentials to (simple) GNN potentials

1. replace pairwise polynomials φ_2 with a simple neural network
2. use multiple rounds of pairwise interaction!

that's the entire core idea.

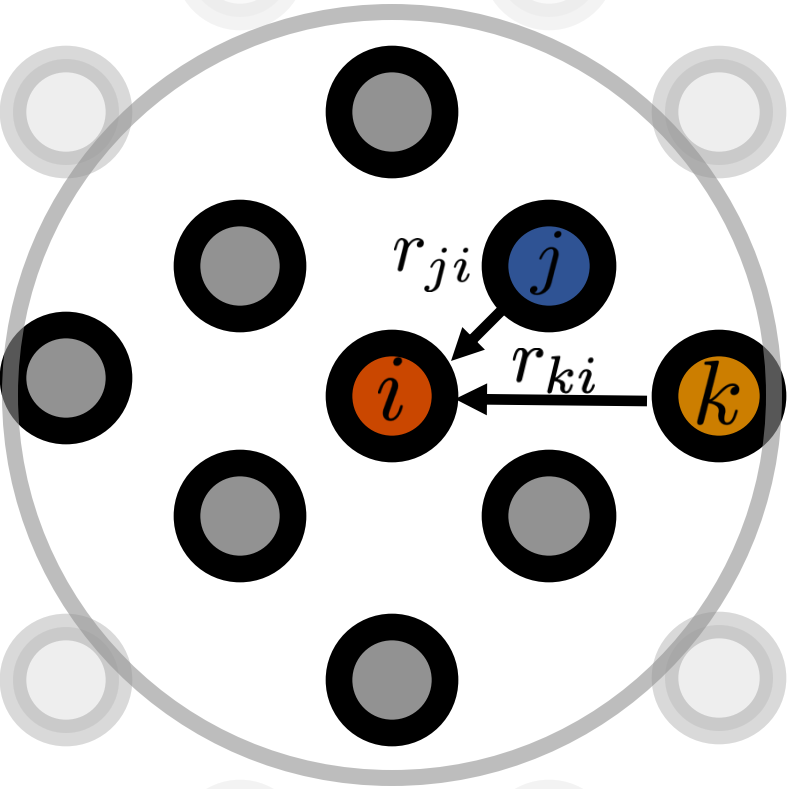
Basic CGCNN interaction: $$\mathbf{x}_i^{l+1} = \sum_j \phi_{ffn}\left([\mathbf{x}_i^l; \mathbf{x}_j^l; b(\mathbf{r}_j - \mathbf{r}_i)]\right)$$

$\mathbf{x}^\ell$	Atom representation
$b(\mathbf{r}_j - \mathbf{r}_i)$	Radial basis
ϕ_{ffn}	Feedforward network
$[a; b]$	Vector concatenation

Variants:
- Atom input featurization
- Spatial basis functions
- Specific neural network architecture
- Incorporation of higher-order functions (angles, dihedrals, …)

Unpacking pair interactions in GNNs

$$W\left([\mathbf{x}_i; \mathbf{x}_j; b(\mathbf{r}_j - \mathbf{r}_i)]\right) = W_d\mathbf{x}_i + W_s\mathbf{x}_j + W_r b(\mathbf{r}_j - \mathbf{r}_i)$$



Expand bond lengths with some basis functions (e.g. Gaussian RBF)

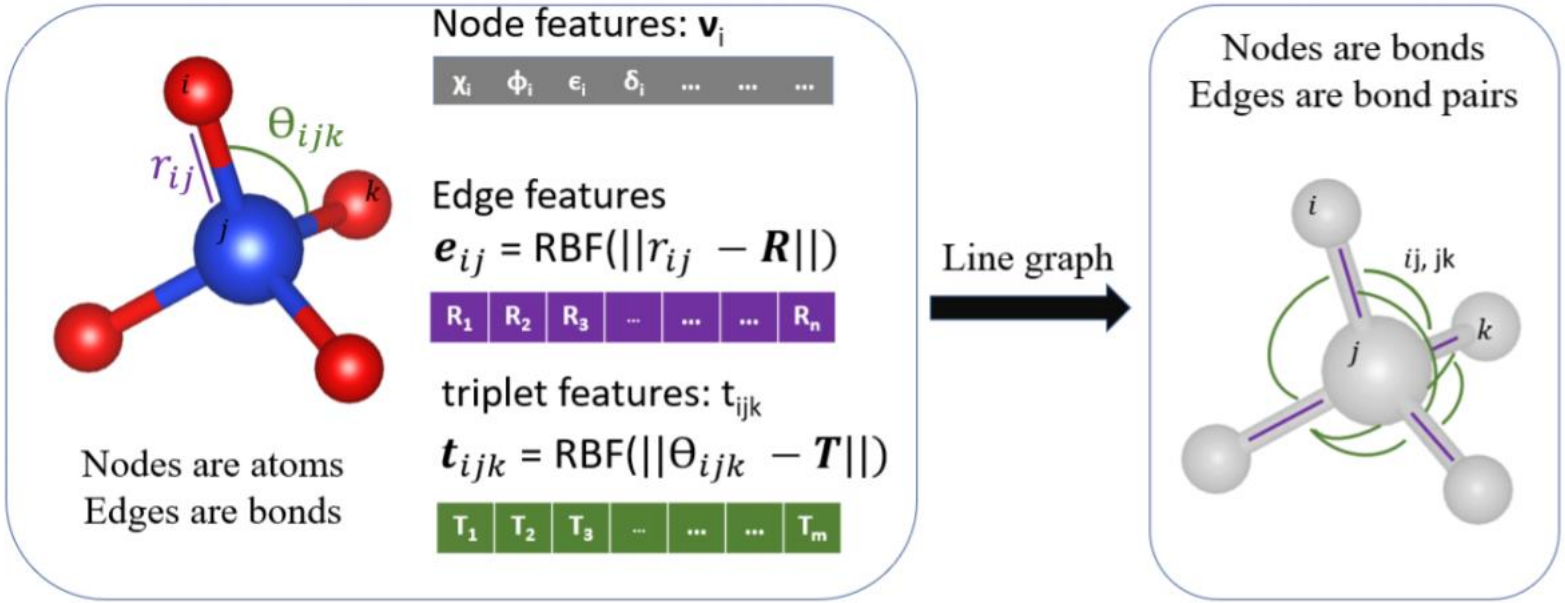
Construct smooth radial functions

Add conditional bias for pair species

$$\beta_{ji} = \exp\left(-\left(r_{ji} - \mathbf{c}\right)/\ell\right)$$

Atomistic Graph & Line Graph

Explicitly represent pairwise and triplet (bond angle) interactions using line graph
Possible to extend for n-body, *e.g.* line graph of line graph



- Graph level prediction, e.g. energy
- Node level predictions, e.g. charges
- Node level derivatives, e.g. forces
- Edge level predictions, e.g. LJ params

nisaba.nist.gov Tesla V100

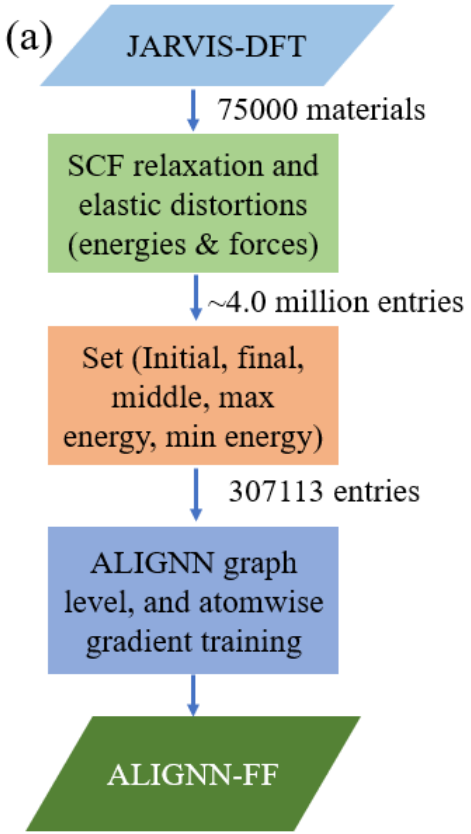
Performance on the JARVIS-DFT Dataset

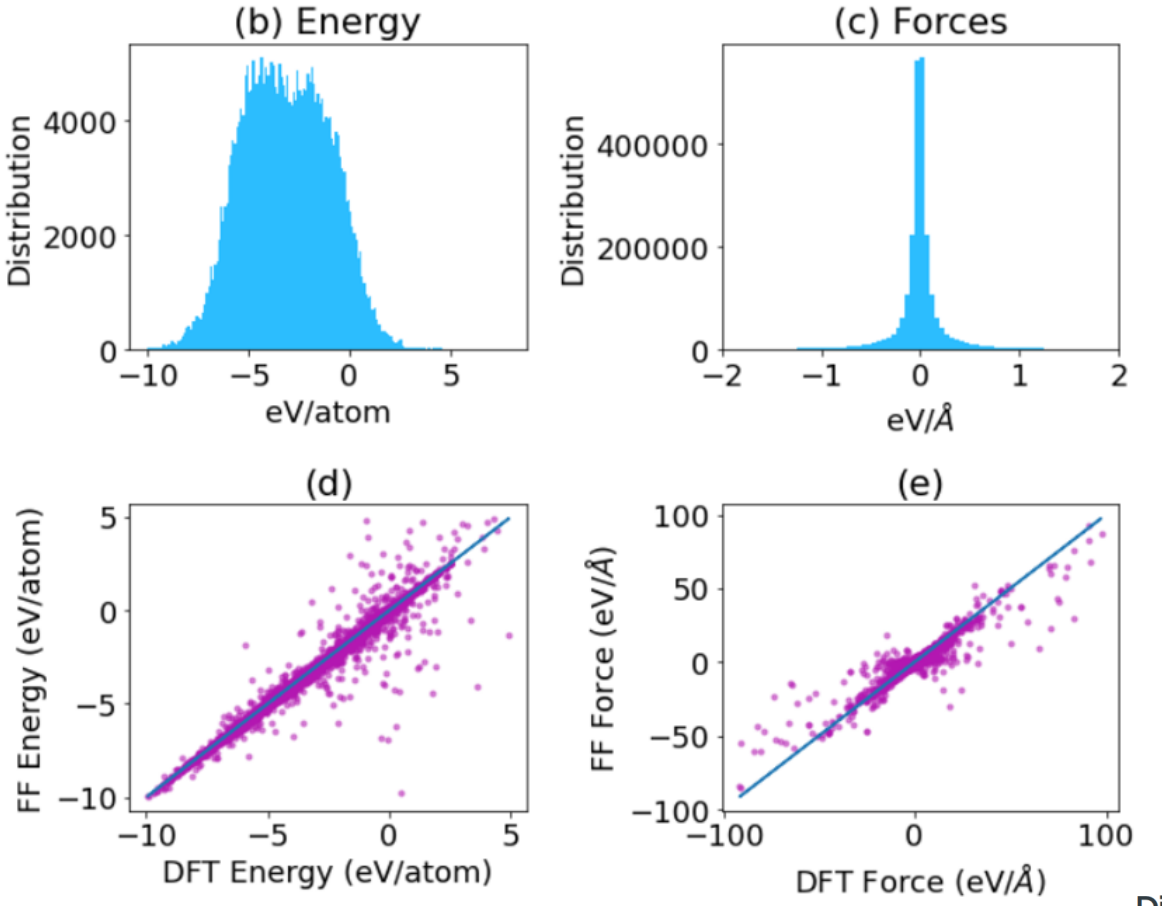
Property	Units	MAD	CFID	CGCNN	ALIGNN	MAD: MAE
Formation energy	eV(atom)$^{-1}$	0.86	0.14	0.063	0.033	26.06
Bandgap (OPT)	eV	0.99	0.30	0.20	0.14	7.07
Total energy	eV(atom)$^{-1}$	1.78	0.24	0.078	0.037	48.11
Ehull	eV	1.14	0.22	0.17	0.076	15.00
Bandgap (MBJ)	eV	1.79	0.53	0.41	0.31	5.77
Kv	GPa	52.80	14.12	14.47	10.40	5.08
Gv	GPa	27.16	11.98	11.75	9.48	2.86
Mag. mom	μB	1.27	0.45	0.37	0.26	4.88
SLME (%)	No unit	10.93	6.22	5.66	4.52	2.42
Spillage	No unit	0.52	0.39	0.40	0.35	1.49
Kpoint-length	Å	17.88	9.68	10.60	9.51	1.88
Plane-wave cutoff	eV	260.4	139.4	151.0	133.8	1.95
ϵ_x (OPT)	No unit	57.40	24.83	27.17	20.40	2.81
ϵ_y (OPT)	No unit	57.54	25.03	26.62	19.99	2.88
ϵ_z (OPT)	No unit	56.03	24.77	25.69	19.57	2.86

Trained on ~55k materials

- Total energy, Formation energy , Ehull
- Bandgap (OPT), Bandgap (MBJ)
- Kv, Gv
- Mag. mom
- ϵx (OPT/MBJ), ϵy (OPT), ϵz (OPT), ϵ (DFPT:elec+ionic)
- Max. piezo. stress coeff (eij)
- Solar-SLME (%)
- Topological-Spillage
- 2D-Exfo. energy
- Kpoint-length
- Plane-wave cutoff
- Max. Electric field gradient
- avg. m_e, avg. m_h
- n-Seebeck, n-PF, p-Seebeck, p-PF

Unified GNN Force-field





Weight	MAE-Energies (eV/atom)	MAE-Forces (eV/Å)
0.1	0.034	0.092
0.5	0.044	0.089
1.0	0.051	0.088
5.0	0.082	0.054
10.0	0.086	0.047

Simulate any combination of 89 elements from the periodic table

$$m_i \frac{\mathrm{d}^2 r_i(t)}{\mathrm{d}t^2} = \sum_j F_{ij}(t) = -\sum_j \nabla_i U\left(r_{ij}(t)\right)$$

Digital Discovery

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal

Check for updates

Cite this: DOI: 10.1039/d2dd00096b

Unified graph neural network force-field for the periodic table: solid state applications

Kamal Choudhary, *ab Brian DeCost, c Lily Major, de Keith Butler, e Jeyan Thiyagalingam e and Francesca Tavazza c

CHIPS-FF: Evaluating Universal MLFFs

ACS MATERIALS LETTERS

Open Access

www.acsmaterialsletters.org

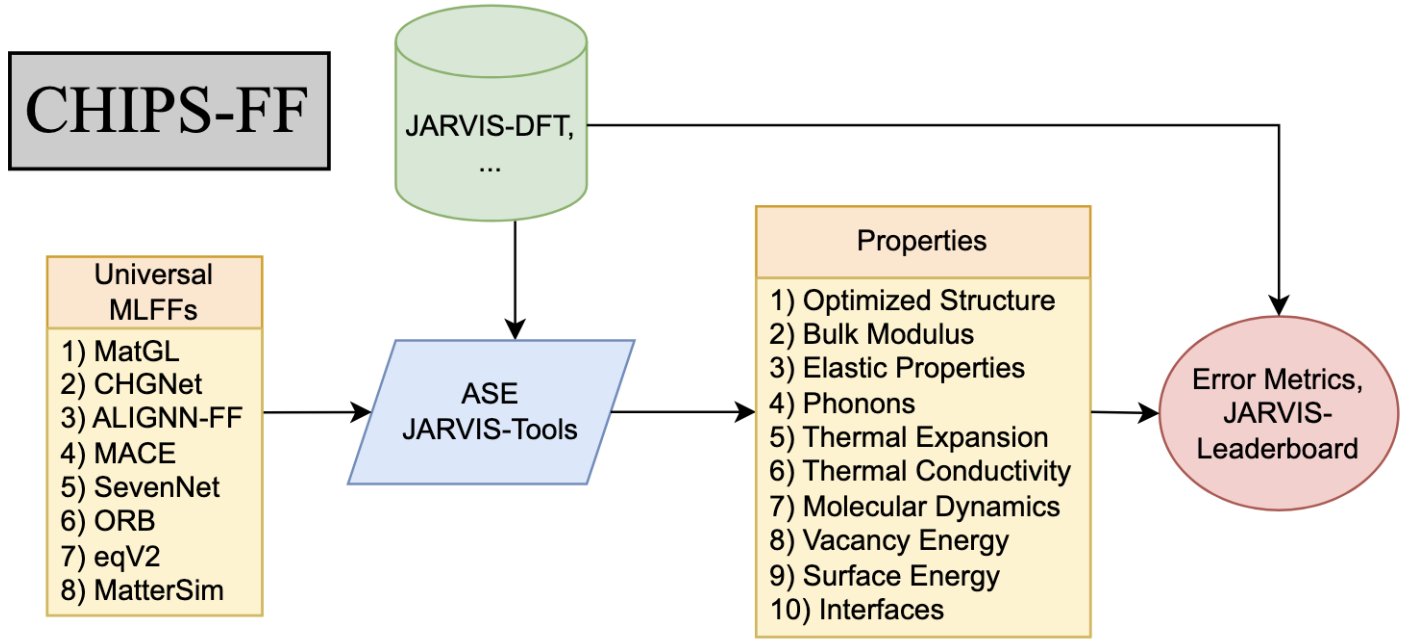
CHIPS-FF: Evaluating Universal Machine Learning Force Fields for Material Properties

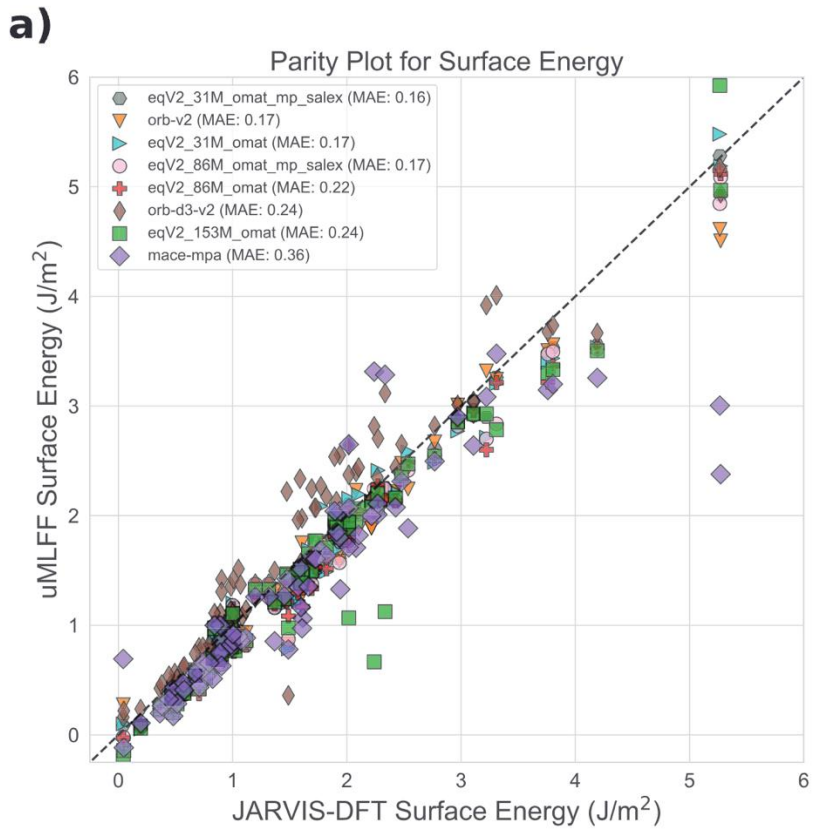
Published as part of ACS Materials Letters special issue "Machine Learning for Materials Chemistry".

Daniel Wines* and Kamal Choudhary

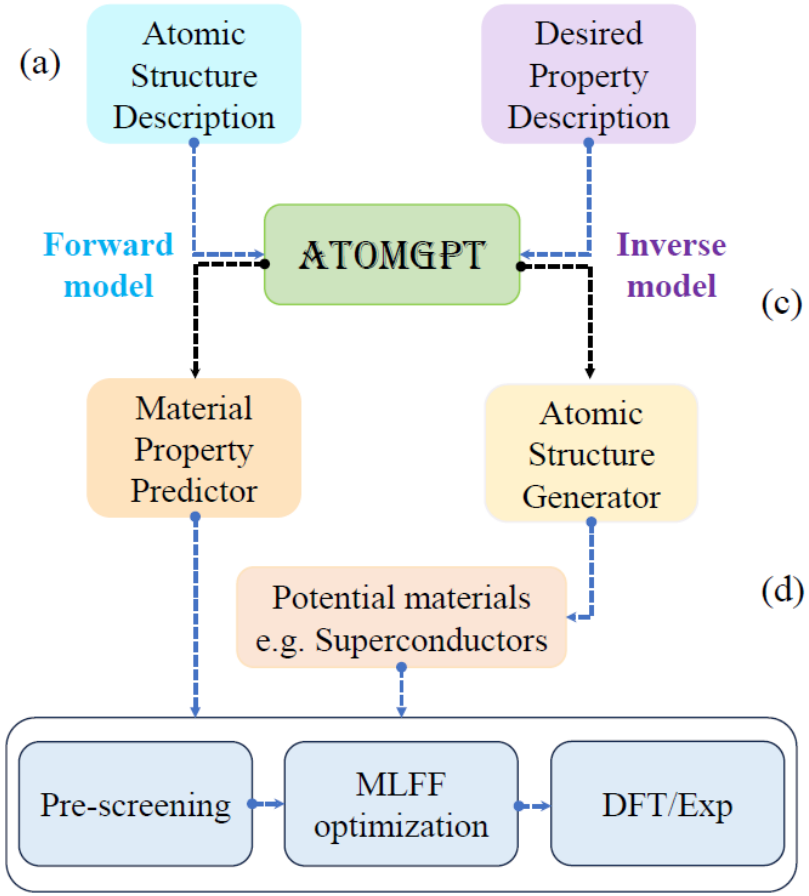
Cite This: *ACS Materials Lett.* 2025, 7, 2105–2114 Read Online

https://github.com/usnistgov/chipsff

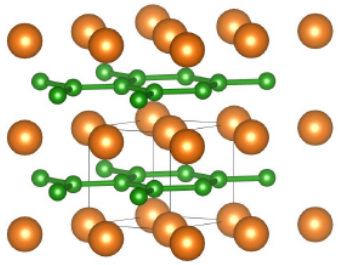




AtomGPT



(b)

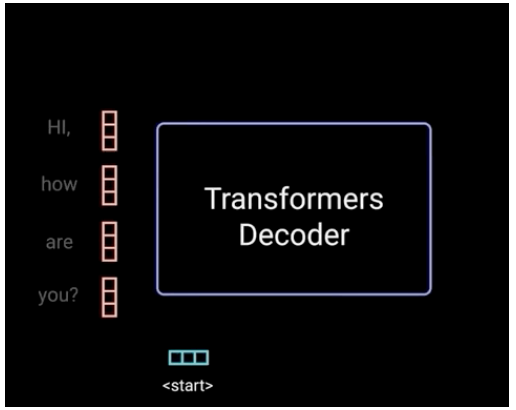


(c)

ChemNLP describer input:
MgB2 is crystallizes in the hexagonal P6/mmm space group. There is one shorter (2.5 Å) and one longer (3.96 Å) B-Mg bond lengths. Mg is bonded in edge-sharing MgB12 Cuboctahedral…
Output: 32.685

(d)

Alpaca input:
{**Instruction**: Below is a description of a superconductor material.,
Input: The chemical formula is MgB_2. The Tc_supercon is 32.685. Generate atomic structure description with lattice lengths, angles, coordinates and atom types.,
Output: 3.07 3.07 3.51\n90 90 120\nMg 0.000 0.000 0.000\nB 0.667 0.333 0.500\nB 0.333 0.667 0.500}

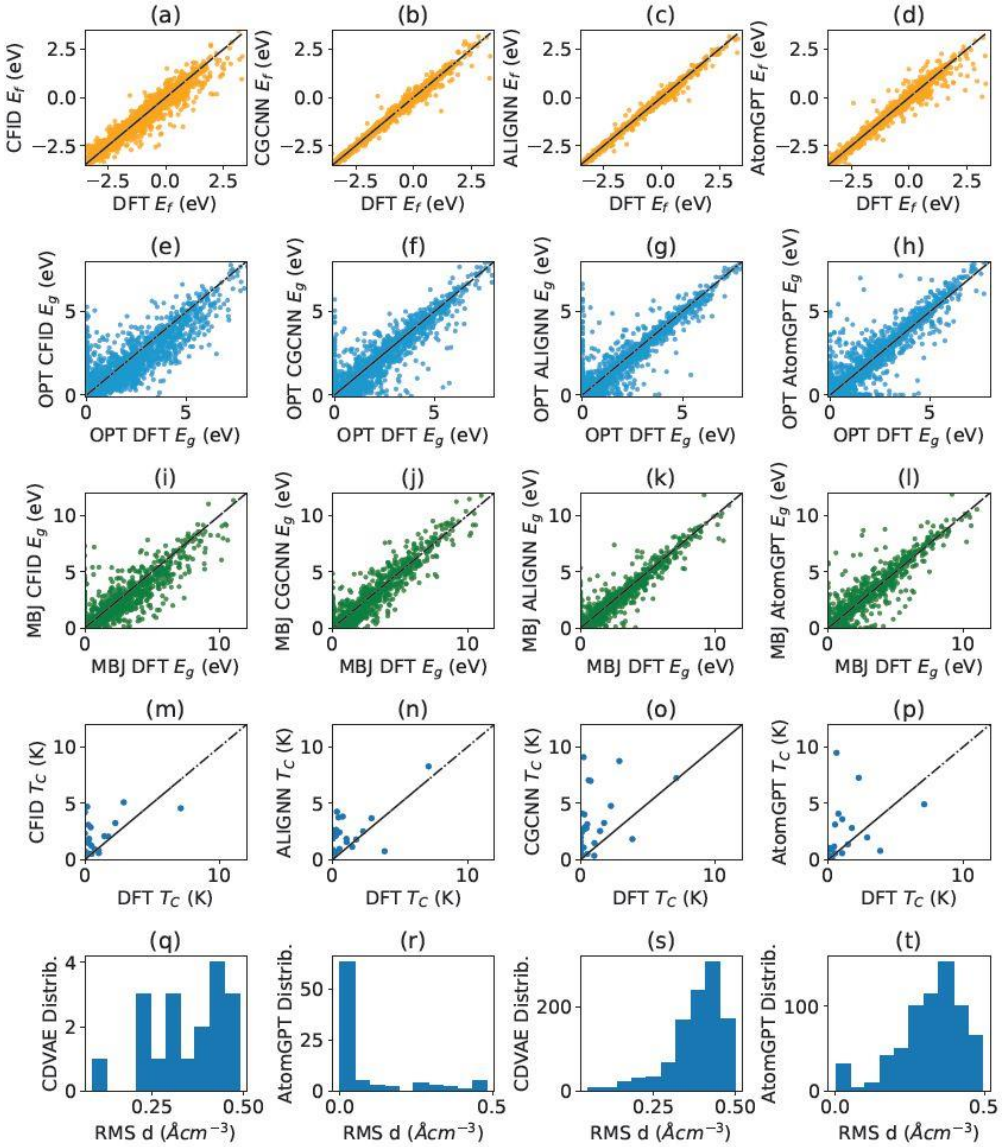


$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

- HuggingFace ecosystem
- Modified language model head for forward models
- Low-rank adaptation (LoRA) for parameter-efficient fine-tuning (PEFT)
- Rotary position embedding (RoPE)
- Transformer reinforcement learning (TRL)

AtomGPT: Atomistic Generative Pretrained Transformer for Forward and Inverse Materials Design, J. Phys. Chem. Lett., 2024

AtomGPT



Forward models				
Prop/MAE	CFID	CGCNN	ALIGNN	AtomGPT
E_{form} (eV/atom)	0.142	0.063	**0.033**	0.072
OPT E_g (eV)	0.299	0.199	0.142	**0.139**
MBJ E_g (eV)	0.531	0.407	**0.310**	0.319
T_C (K)	1.99	2.60	2.03	**1.54**

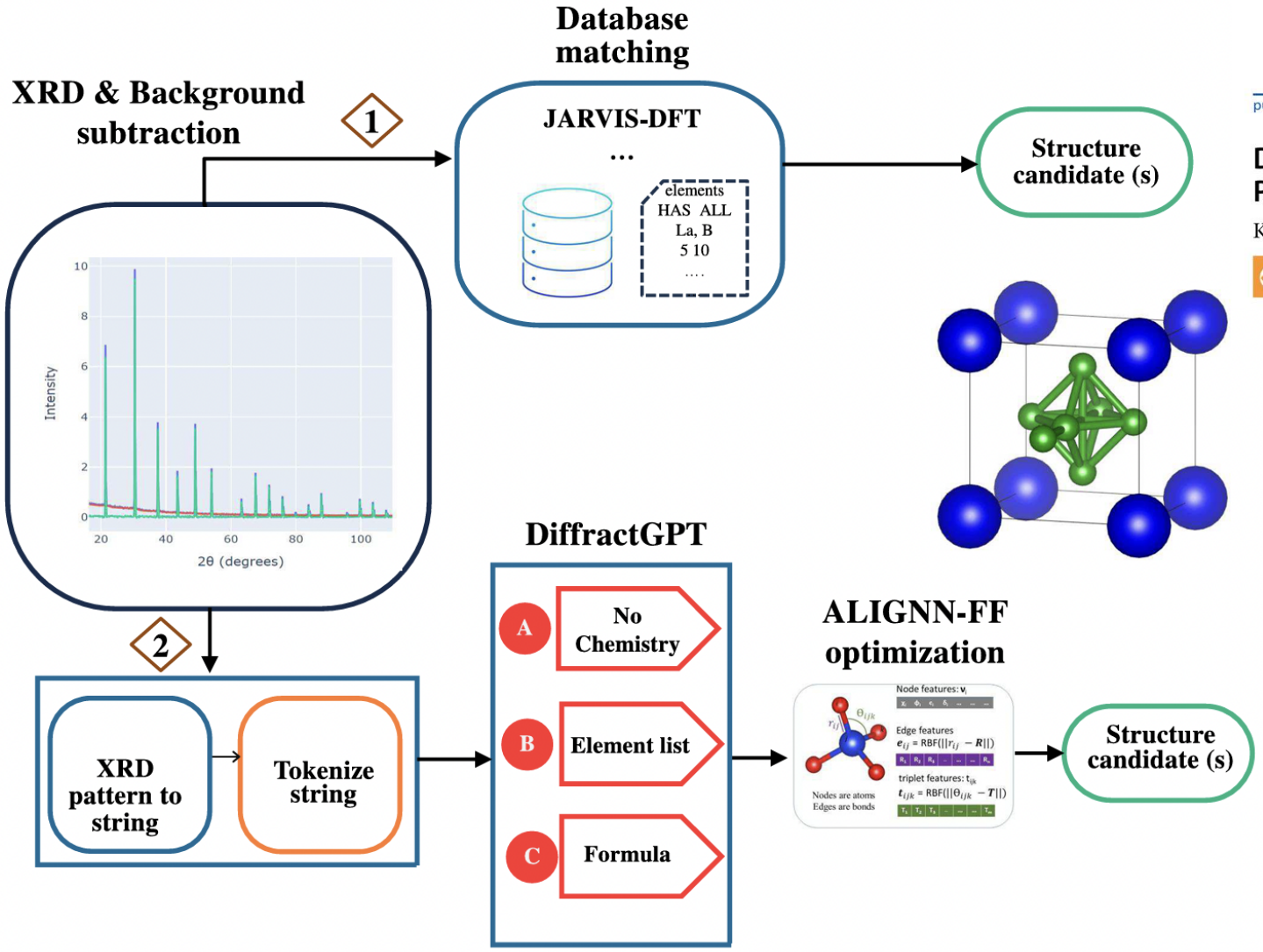
Inverse models		
Database/RMSE	CDVAE	AtomGPT
SuperConDB	0.24	**0.08**
Carbon24	0.36	**0.32**

AtomGPT: Atomistic Generative Pretrained Transformer for Forward and Inverse Materials Design

Kamal Choudhary*

Cite this: *J. Phys. Chem. Lett.* 2024, 15, XXX, 6909−6917
Publication Date: June 27, 2024
https://doi.org/10.1021/acs.jpclett.4c01126

DiffractGPT



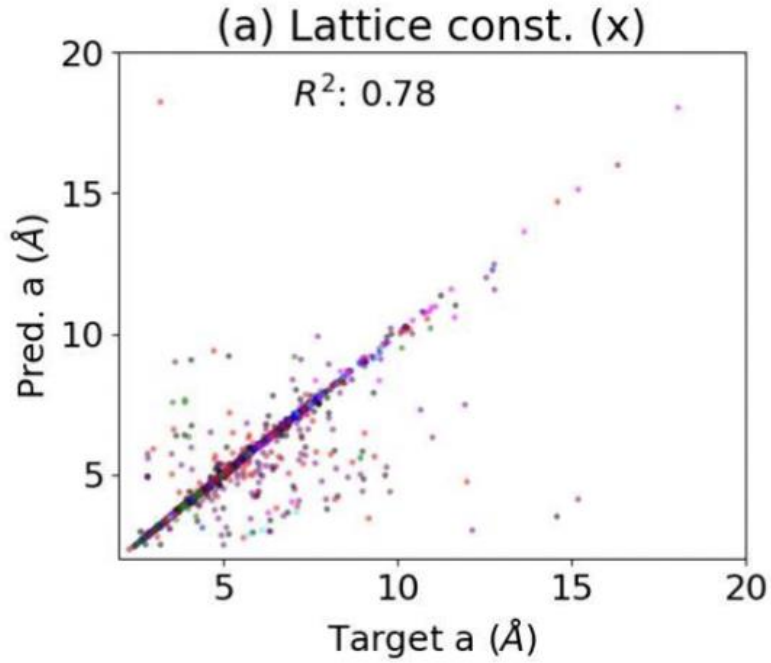
THE JOURNAL OF PHYSICAL CHEMISTRY LETTERS
A JOURNAL OF THE AMERICAN CHEMICAL SOCIETY

pubs.acs.org/JPCL
Letter

DiffractGPT: Atomic Structure Determination from X-ray Diffraction Patterns Using a Generative Pretrained Transformer

Kamal Choudhary*

Cite This: J. Phys. Chem. Lett. 2025, 16, 2110–2119
Read Online



MicroscopyGPT

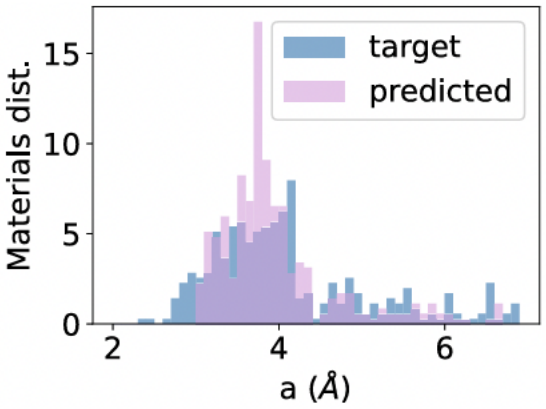
THE JOURNAL OF PHYSICAL CHEMISTRY LETTERS
A JOURNAL OF THE AMERICAN CHEMICAL SOCIETY
Open Access

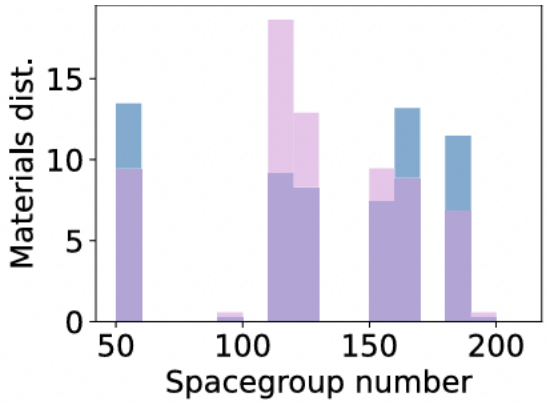
pubs.acs.org/JPCL
Letter

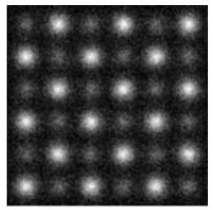
MicroscopyGPT: Generating Atomic-Structure Captions from Microscopy Images of 2D Materials with Vision-Language Transformers

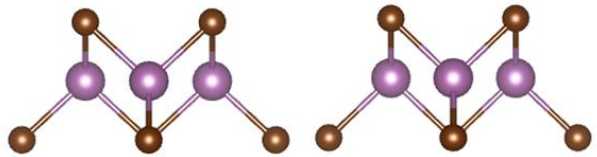
Kamal Choudhary*

Cite This: J. Phys. Chem. Lett. 2025, 16, 7028−7035
Read Online

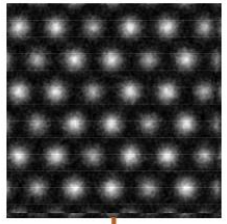








ScBr (P4/nmm)
RMSE: 0.003



AtomVision dataset

{'role': 'user', {'content': [{'type': 'image'}, {'text': 'The chemical formula is Ni_2Si. Generate atomic structure description with lattice lengths, angles, coordinates, and atom types. Also predict the Miller index.'}],}}

Vision Transformer

Text Tokenizer

MicroscopyGPT: Multi-model AtomGPT model for Microscopy Images

{'content': [{'type': 'text', 'text': '\n3.92 3.92 5.02\n90 90 120\nSi 0.667 0.333 0.750\nSi 0.333 0.667 0.250\nNi 0.333 0.667 0.750\nNi 0.667 0.333 0.250\nNi 0.000 0.000 0.500\nNi 0.000 0.000 0.000. The miller index is (0 0 1).'}], 'role': 'assistant'}]}