IMITATION REFINEMENT FOR X-RAY DIFFRACTION SIGNAL PROCESSING

Junwen Bai[1]*, Zihang Lai*[2]*, Runzhe Yang*[3]*, Yexiang Xue*[4]*, John Gregoire*[5]*, Carla Gomes*[1]

[1] Cornell University, Ithaca, NY, USA [2] Oxford University, Oxford, UK
[3] Princeton University, Princeton, NJ, USA [4] Purdue University,West Lafayette, IN, USA
[5] California Institute of Technology, Pasadena, CA, USA

ABSTRACT

Many real-world tasks involve identifying signals from data satisfying background or prior knowledge. In domains like materials discovery, due to the flaws and biases in raw experimental data, the identification of X-ray diffraction (XRD) signals often requires significant (manual) expert work to find refined signals that are similar to the *ideal* theoretical ones. Automatically refining the raw XRD signals utilizing simulated theoretical data is thus desirable. We propose *imitation refinement*, a novel approach to refine imperfect input signals, guided by a pre-trained classifier incorporating prior knowledge from simulated theoretical data, such that the refined signals *imitate* the *ideal* ones. The *classifier* is trained on the *ideal* simulated data to classify signals and learns an embedding space where each class is represented by a *prototype*. The *refiner* learns to refine the imperfect signals with small modifications, such that their embeddings are closer to the corresponding prototypes. We show that the refiner can be trained in both supervised and unsupervised fashions. We further illustrate the effectiveness of the proposed approach both qualitatively and quantitatively in an X-ray diffraction signal refinement task in materials discovery.

Index Terms— refinement, classification, X-ray diffraction signals, embeddings, neural nets

1. INTRODUCTION

Many real-world tasks involve identifying meaningful signals satisfying background or prior knowledge from limited amount of labeled data [1]. Furthermore, the raw data are often corrupted with noise [2], which makes it even harder to identify meaningful signals. On the other hand, in many domains like scientific discovery, though the experimental data might be flawed or biased, *ideal* data can often be synthesized easily [3, 4]. It is thus desirable to incorporate knowledge from *ideal* data to refine the quality of the raw signals to make them more meaningful and recognizable.

For instance, in materials discovery, where we would like to discover new materials, each material is composed of one or more crystal structures (also simply called phases), where each phase is distinguished by a unique X-ray diffraction (XRD) signal. Materials scientists are interested in not only predicting the properties of materials [5], but also identifying their crystal structures or phases, which should be similar to (ideal) theoretical phases [6]. Phase identification is challenging because the raw phases from experiments are often mixed with each other and further corrupted with noise. Currently it is mainly a manual task, a key bottleneck of the materials discovery process.

In related work, data denoising and restoration methods [7, 8, 9] often require paired clean and noisy data for training, which is not available in our setting, since only theoretical (simulated) and noisy experimental data are given and the correspondences are unknown. Similarly, style transfer[10], also requires paired images or cycle consistency. Recently, GAN-based domain adaptation models [11, 12, 13] bring a new perspective to reduce the need for the correspondence supervision. But these methods mainly focus on prediction instead of refining the quality of the imperfect input signals.

We propose a novel approach called **imitation refinement**, which improves the quality of imperfect signals by **imitating** ideal signals, guided by a classifier with prior knowledge pre-trained on the ideal dataset. The classifier incorporates such knowledge by learning a meaningful embedding space. Imitation refinement applies small modifications to the imperfect signals such that (1) the refined signals have better quality and are similar to the ideal signals and (2) the pre-trained classifier can achieve better classification accuracy on the refined signals. We show that both objectives can be achieved even with limited amount of data.

The main contribution of our work is to provide a novel framework for imitation refinement, which can be used to improve the quality of imperfect signals under the supervision from a classifier containing prior knowledge. Our second contribution is to find an effective way to incorporate the prior knowledge from the ideal data into the classifier. The third contribution of this work is to provide a way to train the refiner even if the imperfect inputs have no supervision.

Using a materials discovery dataset, we show that for the imperfect input experimental phases, the refined phases are closer to the quantum-mechanically computed ideal phases. In addition, we achieve higher classification accuracy on the refined phases. We show that even in the unsupervised case,

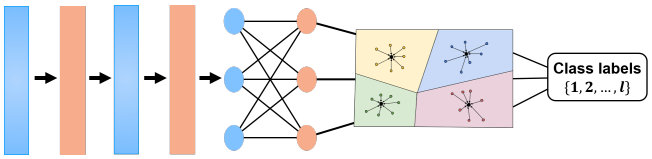


Fig. 1: Prototypical classifier not only predicts labels, but also learn a meaningful embedding space. The center of the cluster for each class is called prototype.

the refinement can help improve the quality of input signals.

2. IMITATION REFINEMENT

2.1. Notation

In imitation refinement, we are given an ideal dataset $\mathcal{D}_{ideal} = \{(x_i^{ideal}, y_i^{ideal})\}_{i=1}^{N}$. Each ideal d-dimensional feature $x_i^{ideal} \in \mathcal{X}^{ideal} \subseteq \mathbb{R}^d$ is a realization from a random variable X^{ideal}, and the label $y_i^{ideal} \in \mathcal{Y}$, where $\mathcal{Y}$ is a discrete set of classes $\{0, 1, ..., l\}$ in this problem. In addition, we are also given the imperfect training data $\mathcal{D}_{imp}$. In the supervised/targeted case, $\mathcal{D}_{imp} = \{(x_i^{imp}, y_i^{imp})\}_{i=1}^{M}$ where $x_i^{imp} \in \mathcal{X}^{imp} \subseteq \mathbb{R}^d$ is a realization of a random variable X^{imp} and the labels $y_i^{imp} \in \mathcal{Y}$. In the unsupervised/non-targeted case, $\mathcal{D}_{imp} = \{x_i^{imp}\}_{i=1}^{M}$ where $x^{imp} \in \mathcal{X}^{imp}$ and the labels are not available. We assume there is a slight difference [14, 15] between $\mathcal{X}^{imp}$ and $\mathcal{X}^{ideal}$.

2.2. Problem Description

Our goal is to learn a function $\mathcal{R} : \mathcal{X}^{imp} \to \mathcal{X}^{rfd}$, where $\mathcal{X}^{rfd} \subseteq \mathcal{X}^{ideal}$, that refines the imperfect signals into ideal signals (e.g. the theoretically computed corresponding signals), with the guidance from a pre-trained classifier $\mathcal{C}$. $\mathcal{C}$ is the composition $\mathcal{G}_\psi \circ \mathcal{F}_\theta$ where $\mathcal{F}_\theta : \mathcal{X}^{ideal} \to \mathbb{R}^m$ is an embedding function (m-dimensional embedding space) and $\mathcal{G}_\psi : \mathbb{R}^m \to \mathcal{Y}$ is a prediction function. For the inputs x, we hope $\mathcal{C}(\mathcal{R}(x))$ can give better results than $\mathcal{C}(x)$, and $\mathcal{R}(x)$ has better quality than x, by imitating the signals in $\mathcal{X}^{ideal}$.

2.3. Pre-trained Prototypical Classifier

Inspired by recently proposed prototypical networks [16], the classifier is trained to learn a meaningful embedding space to better incorporate the prior knowledge as well as the class prediction, where each class can be represented by a prototype embedding and embeddings from each class form a cluster surrounding the prototype. The classifier is thus called *prototypical classifier*.

To learn such a prototypical classifier, besides the class prediction loss given by a classification loss $\mathcal{L}_\mathcal{C}(\theta, \psi) = \sum_i \ell(\mathcal{G}_\psi(\mathcal{F}_\theta(x_i^{ideal})), y_i^{ideal})$ where $\ell(\cdot)$ can be the cross entropy loss $\mathcal{H}(\cdot)$ or other supervised losses, we further add a loss defining the distances to the ground-truth prototypes in the embedding space given a distance function $\delta : \mathbb{R}^m \times \mathbb{R}^m \to [0, +\infty)$:

$$\mathcal{L}_\mathcal{F}(\theta) = \sum_i -\log \frac{\exp(-\delta(\mathcal{F}_\theta(x_i^{ideal}), c_{y_i^{ideal}}))}{\sum_{k' \in \mathcal{Y}} \exp(-\delta(\mathcal{F}_\theta(x_i^{ideal}), c_{k'}))} \quad (1)$$

Fig. 1 gives an overview of the prototypical classifier. In each training step, the batch of samples are randomly selected from each class to ensure each class has a least one sample. The prototype of each class is randomly initialized and updated by the mean of the embeddings from the class and the prototype from last batch. More specifically, Prototypes are updated through $c_k^t \leftarrow \frac{1}{|\mathcal{D}_k^{ideal}|+1}(c_k^{t-1} + \sum_{y_i^{ideal}=k} \mathcal{F}_\theta(x_i^{ideal}))$ where $\mathcal{D}_k^{ideal}$ is the subset of $\mathcal{D}_{ideal}$ containing all the samples from class k. The batch loss is computed via $\mathcal{L}_{batch}(\theta, \psi) \leftarrow \mathcal{L}_\mathcal{C}(\theta, \psi) + \lambda \mathcal{L}_\mathcal{F}(\theta)$. Then we can update the parameters θ, ψ by taking an Adam[17] step on the batch loss.

Algorithm 1 One epoch in the training for the refiner when labels are available

Input: Imperfect training dataset $\mathcal{D}_{imp} = \{(x_i^{imp}, y_i^{imp})\}_{i=1}^{M}$ where $y_i^{imp} \in \{1, ..., l\}$, max number of batches (T), the batch size (N_c), the prototypical classifier $\mathcal{C}$ and the prototypes c_k.
Output: Refiner $\mathcal{R}_\phi$.

1: **for** t=1..T **do**
2: Sample N_c samples from training set D_{imp} : $\{x_i, y_i\}_{i=1}^{N_c}$.
3: Let $r_i = \mathcal{R}(x_i)$ be the refined inputs.
4: Let $e_i = \mathcal{F}(r_i)$ be the embedded refined inputs.
5: Let $c_i = \mathcal{G}(e_i)$ be the predicted labels.
6: Compute $\mathcal{L}_\mathcal{R}(\phi)$ in equation (2)
- $\ell_{\text{pred}} = \frac{1}{N_c} \sum_i \mathcal{H}(c_i, y_i)$
- $\ell_{\text{proto}} = \frac{1}{N_c} \sum_i -\log \frac{\exp(-\delta(e_i, c_{y_i}))}{\sum_{k'} \exp(-\delta(e_i, c_{k'}))}$
- $\ell_{\text{reg}} = \frac{1}{N_c} \sum_i ||\Psi(r_i) - \Psi(x_i)||_p$

7: Update parameters ϕ through back-propagation based on the loss $\mathcal{L}_\mathcal{R}(\phi)$.

2.4. Imitation Refiner

The pre-trained prototypical classifier is then applied to guide the training of the refiner $\mathcal{R}_\phi : \mathcal{X}^{imp} \to \mathcal{X}^{rfd}$ with learnable parameters ϕ. We propose to learn ϕ by minimizing a combination of three losses:

$$\mathcal{L}_R(\phi) = \sum_i (\ell_{\text{pred}}(\phi; x_i^{imp}, \mathcal{Y}) + \alpha \ell_{\text{reg}}(\phi; x_i^{imp}) + \beta \ell_{\text{proto}}(\phi; x_i^{imp}, \mathcal{F}(\mathcal{X}^{ideal}))). \quad (2)$$

where x_i^{imp} is the i^{th} imperfect training sample and $\mathcal{F}(\mathcal{X}^{ideal})$ is the embedding space formed by the embeddings of samples from the space $\mathcal{X}^{ideal}$. α and β are coefficients that trade

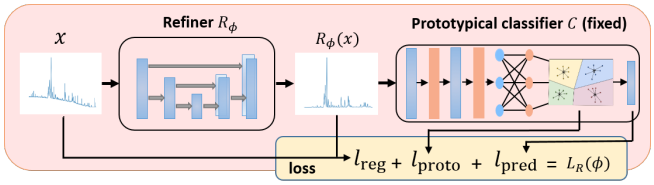


Fig. 2: The refiner $\mathcal{R}_\phi$ is trained in an end-to-end fashion. The pre-trained classifier $\mathcal{C}$ provides ℓ_{proto} and ℓ_{pred}. Loss ℓ_{reg} is given by the difference between the refined input and the raw input in either the raw space or some feature space.

off different losses. They are set to 0.01 and 1 respectively by grid search on them. Note that once the classifier $\mathcal{C}$ is trained, it is fixed along with the prototypes c_k's during the training of the refiner. Furthermore, $\mathcal{C}$ provides loss functions (ℓ_{pred} and ℓ_{proto}) to the training of $\mathcal{R}_\phi$. The refiner can be trained in both targeted and non-targeted fashions, depending on whether the labels of the imperfect training samples are provided or not. In the targeted case, prediction loss ℓ_{pred} is the loss given by the difference between the predicted labels of the refined input signals and the ground-truth labels:

$$\ell_{\text{pred}}(\phi; x_i^{imp}, \mathcal{Y}) = \mathcal{H}(\mathcal{C}(\mathcal{R}_\phi(x_i^{imp})), y_i^{imp}) \quad (3)$$

where $\mathcal{H}$ is the cross-entropy loss. In the non-targeted case, we simply change the cross-entropy loss to the entropy loss, $\mathcal{H}(\mathcal{C}(\mathcal{R}_\phi(x_i^{imp})))$, to represent the uncertainty of the classifier $\mathcal{C}$ on the refined signals. The goal is to minimize the entropy to force the refiner to learn more meaningful refined signals which could be better recognized by $\mathcal{C}$.

We further introduce the prototypical loss ℓ_{proto} to further guide the refinement towards the corresponding prototypes in the embedding space, which is more robust. In the targeted case, the prototypical loss is given by the negative log-likelihood on the distances between the embeddings of the refined signals and the ground-truth prototypes:

$$\begin{aligned} &\ell_{\text{proto}}(\phi; x_i^{imp}, \mathcal{F}(\mathcal{X}^{ideal})) \\ &= -\log \frac{\exp(-\delta(\mathcal{F}(\mathcal{R}_\phi(x_i^{imp})), c_{y_i^{imp}}))}{\sum_{k'} \exp(-\delta(\mathcal{F}(\mathcal{R}_\phi(x_i^{imp})), c_{k'}))} \end{aligned} \quad (4)$$

In the non-targeted case, we use entropy loss:

$$\ell_{\text{proto}}(\phi; x_i^{imp}, \mathcal{F}(\mathcal{X}^{ideal})) = \sum_{k=1}^{l} -p_{i,k}^{imp} \log p_{i,k}^{imp} \quad (5)$$

where $p_{i,k}^{imp} = \frac{\exp(-\delta(\mathcal{F}(\mathcal{R}_\phi(x_i^{imp})), c_k))}{\sum_{k'} \exp(-\delta(\mathcal{F}(\mathcal{R}_\phi(x_i^{imp})), c_{k'}))}$.

Note that for an imperfect sample x_i^{imp}, we are looking for an ideal signal in $\mathcal{X}^{ideal}$ that is most related to x_i^{imp}. $\mathcal{R}$ should modify the input as little as possible to remain the contents in the imperfect input samples [10]. Thus, we introduce the loss ℓ_{reg} to regularize the changes made for the input:

$$\ell_{\text{reg}}(\phi; x_i^{imp}) = ||\Psi(\mathcal{R}(x_i^{imp})) - \Psi(x_i^{imp})||_p \quad (6)$$

where $||\cdot||_p$ is p-norm and Ψ maps the raw input into a feature space. Ψ can be an identity map $\Psi(x) = x$ or more abstract features such as the feature maps after the first or second convolution layer. This loss works for both targeted and non-targeted cases since it does not rely on the labels. Such regularization can also help avoid learning an ill-posed mapping from $\mathcal{X}^{imp}$ to $\mathcal{X}^{ideal}$ such as a many-to-one mapping that maps all the imperfect signals from class k to one ideal signal in class k regardless the raw contents in the imperfect signals. This mapping could achieve very small ℓ_{pred} and ℓ_{proto} but would result in a trivial mapping.

The refiner $\mathcal{R}$ is trained in an end-to-end way as described above and all the parameters are updated through back-propagation. The pseudocode to train the refiner is provided in Algorithm 1 in the targeted case. In the non-targeted case, the algorithm is simply replacing the targeted losses with non-targeted losses as described. The overall structure of the refiner is shown in Fig. 2.

3. EXPERIMENTS

High-throughput combinatorial materials discovery is a materials science task whose intent is to discover new materials using a variety of methods including X-ray diffraction signal analysis[18]. The raw imperfect X-ray diffraction signals (XRD) from experiments are often unsatisfiable because the data corruption could happen in any step of the data processing. In this work, we show how imitation refinement (IR) can help refine XRDs to approach the ideal signals from computational models to validate that useful domain knowledge is learned by the pre-trained classifier. We show our performance via two metrics. First, the refined XRDs can achieve better classification accuracy even if the classifier is fixed. Second, we directly show the improvement of the quality of the refined XRDs both qualitatively and quantitatively. We measure the difference between the ground-truth XRDs and refined XRDs on ℓ_1 loss, ℓ_2 loss, KL-divergence and cross correlation. Qualitative results are also shown in Fig.4.

Dataset: The dataset used in this application is from the Materials Project [19]. The ideal simulated data have approximately 240,000 samples from 7 classes. The imperfect dataset has only 1,494 experimental samples from 7 classes. Both datasets are divided into training and test data according to the ratio 9:1 for the classifier and refiner respectively.

Implementation details: The refiner network, $\mathcal{R}_\phi$, is U-Net [20]. For the classifier $\mathcal{C}$, we use two structures DenseNet [21] and VGG [22] to show that the imitation refinement framework works for different classifiers.

Classification results: The baselines are given by the classifiers pre-trained on the ideal dataset tested on the imperfect data. The accuracies from standard and prototypical classifiers based on VGG-19 model are 68.54% and 69.01% respectively. Similarly, classifiers based on DenseNet model give 67.74% and 70.82%. Table 1 presents the label predic-

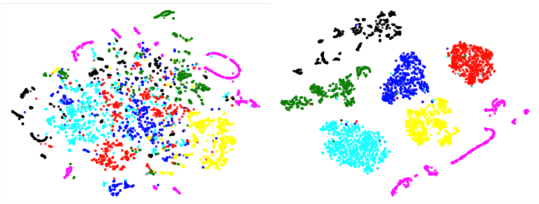
model	accuracy
DWT	71.28%
ADDA	73.78%
GTA	73.18%

IR	VGG	T	73.98%
		NT	71.63%
	DenseNet	T	**80.05%**
		NT	74.74%

Table 1: Different accuracies from different methods including our Imitation Refinement ("IR"). We give the results in both targeted ("T") and non-targeted ("NT") cases. Our method outperforms other methods.

Models	ℓ_{proto}	ℓ_{pred}	ℓ_{reg}	Accuracy
UNet+DenseNet	Y	N	Y	79.33%
UNet+DenseNet	N	Y	Y	75.76%
UNet+DenseNet	Y	Y	Y	**80.05%**

Table 2: The combination of 3 losses gives the best accuracy. "Y" stands for yes and "N" stands for no.



(a) GTA embedding space (b) Our embedding space

Fig. 3: TSNE visualization of the embedding spaces learned by GTA and imitation refinement. Each color denotes a class.

tion accuracies from different methods or settings. Discrete wavelet transform [23] is a widely used signal denoising technique in materials science domain. ADDA [12] and GTA [13] are recently proposed adversarial domain adaptation method aiming at learning different feature extraction networks for two similar domains. Both use DenseNet as the classifier. All the methods use the same amount of supervision (data and labels from the training data of ideal and imperfect datasets). Further, in the non-targeted cases, the supervision from the imperfect training data would not be given.

Ablation study: To show the combination of the different losses is necessary and meaningful, we show the results in Table 2 when either ℓ_{proto} or ℓ_{pred} is ablated. Note that all the reported numbers are averaged over 5 independent runs.

Quantitative results: Regarding the quality of refined XRDs: We directly measure the differences between refined XRDs and the ground-truth theoretical XRDs on 4 metrics, ℓ_1, ℓ_2, KL-divergence and cross correlation. We compute medians of the differences over all the test data (Table 3). Imitation refinement outperforms other methods. We also present the embedding spaces learned by GTA and IR (Fig. 3).

Qualitative analysis: As shown in Fig.3, IR learns more meaningful embedding space. Fig.4 gives two randomly chosen examples of the raw XRD, refined XRD and the ground-truth XRD for materials $NbGa_3$ and Mn_4Al_{11}.

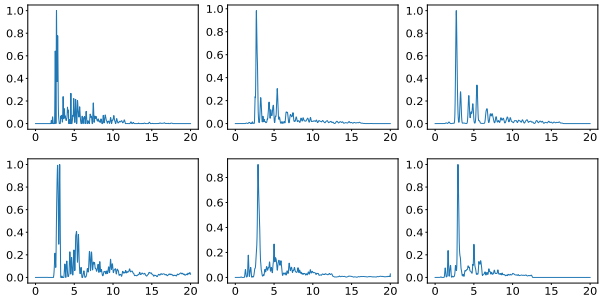


Fig. 4: The visualization of the raw, refined and ground-truth XRDs for $NbGa_3$ and Mn_4Al_{11} (randomly chosen). We can see imitation refinement can add missing peaks, remove redundant peak and denoise the XRD pattern from learning.

Models	ℓ_1	ℓ_2	KL	NCC
Raw XRD	26.749	1.718	12.195	20.870
DWT	25.642	1.706	11.972	20.884
GTA	90.914	6.861	23.306	20.618
IR-VGG	26.200	**1.603**	**11.686**	20.956
NT IR-VGG	26.034	1.630	12.545	21.232
IR-DenseNet	**25.101**	1.671	11.945	**22.481**
NT IR-DenseNet	27.235	1.754	11.834	21.079

Table 3: Differences between the refined XRDs and the ground-truth XRDs on metrics ℓ_1, ℓ_2, KL and normalized cross correlation (NCC). The difference between the raw XRDs and the ground-truth XRDs gives the baseline. The differences shown in the table are the medians over all the test data. For ℓ_1, ℓ_2, KL, the smaller the better. For NCC, the larger, the better. The best results are shown in **bold**.

4. CONCLUSIONS

Imitation refinement improves the quality of imperfect signals by imitating *ideal signals*. Using the prior knowledge captured by a *prototypical classifier* trained on an ideal dataset, a refiner learns to apply modifications to imperfect signals to improve their qualities. A general end-to-end neural framework is proposed to address this refinement task and gives promising results in an XRD signal refinement task and refines the XRDs to be closer to the ground-truth. This work has a potential to save lots of manual work for material scientists. We also show that imitation refinement could work even if labels are not provided. Imitation refinement is adaptable to other similar situations in scientific discovery in chemistry, physics, etc. We hope our work will stimulate additional imitation refinement efforts.

Acknowledgments. Work supported by an NSF Expedition award for Computational Sustainability (CCF-1522054), NSF Computing Research Infrastructure (CNS-1059284), NSF Inspire (1344201), a MURI/AFOSR grant (FA9550), and a grant from the Toyota Research Institute.

5. REFERENCES

[1] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien, "Semi-supervised learning," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542–542, 2009.

[2] Jan Steinbrener, Johanna Nelson, Xiaojing Huang, Stefano Marchesini, David Shapiro, Joshua J. Turner, and Chris Jacobsen, "Data preparation and evaluation techniques for x-ray diffraction microscopy," *Optics express*, vol. 18, no. 18, pp. 18598–18614, 2010.

[3] Donald B. Rubin, "Discussion statistical disclosure limitation," *Journal of official Statistics*, vol. 9, no. 2, pp. 461, 1993.

[4] Ronan Le Bras, Richard Bernstein, John M. Suram, Santosh K. Gregoire, Carla P. Gomes, Bart Selman, and R. Bruce van Dover, "A computational challenge problem in materials discovery: Synthetic problem generator and real-world datasets," *Proceedings of the 28th international conference on machine learning*, 2014.

[5] Woon Bae Park, Jiyong Chung, Jaeyoung Jung, Keemin Sohn, Satendra Pal Singh, Myoungho Pyo, Namsoo Shin, and K-S Sohn, "Classification of crystal structure using a convolutional neural network," *IUCrJ*, vol. 4, no. 4, pp. 486–494, 2017.

[6] Scott A. Speakman, "Introduction to x-ray powder diffraction data analysis," *Center for Materials Science and Engineering at MIT*, 2013.

[7] Junyuan Xie, Linli Xu, and Enhong Chen, "Image denoising and inpainting with deep neural networks," in *Advances in neural information processing systems*, 2012, pp. 341–349.

[8] Sungmin Cha and Taesup Moon, "Neural adaptive image denoiser," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2981–2985.

[9] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang, "Learning a deep convolutional network for image super-resolution," in *European Conference on Computer Vision*. Springer, 2014.

[10] Leon A Gatys, Alexander S Ecker, and Matthias Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2414–2423.

[11] Zhong Meng, Jinyu Li, Yifan Gong, and Biing-Hwang Juang, "Adversarial teacher-student learning for unsupervised domain adaptation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5949–5953.

[12] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell, "Adversarial discriminative domain adaptation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[13] Swami Sankaranarayanan and Yogesh Balaji, "Generate to adapt: Aligning domains using generative adversarial networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[14] Hidetoshi Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of statistical planning and inference*, vol. 90, no. 2, pp. 227–244, 2000.

[15] Masashi Sugiyama and Motoaki Kawanabe, *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*, MIT Press, 2012.

[16] Jake Snell, Kevin Swersky, and Richard Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 4077–4087.

[17] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[18] Martin L. Green, C.L. Choi, J.R. Hattrick-Simpers, AM Joshi, I. Takeuchi, S.C. Barron, E. Campo, T. Chiang, S. Empedocles, J.M. Gregoire, et al., "Fulfilling the promise of the materials genome initiative with high-throughput experimental methodologies," *Applied Physics Reviews*, vol. 4, no. 1, pp. 011105, 2017.

[19] Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William D. Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, et al., "Commentary: The materials project: A materials genome approach to accelerating materials innovation," *Apl Materials*, vol. 1, no. 1, pp. 011002, 2013.

[20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.

[21] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger, "Densely connected convolutional networks," 2017.

[22] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[23] Chunsheng Cai and Peter de B. Harrington, "Different discrete wavelet transforms applied to denoising analytical data," *Journal of chemical information and computer sciences*, vol. 38, no. 6, pp. 1161–1170, 1998.