Transfer learning application of self-supervised learning in ARPES

Sandy Adhitia Ekahana\textsuperscript{a}\textsuperscript{*}, Genta Indra Winata\textsuperscript{a,b}, Y Soh\textsuperscript{a}, Anna Tamai\textsuperscript{c}, Radovic Milan\textsuperscript{d} and Ming Shi\textsuperscript{e}

\textsuperscript{1} Paul Scherrer Institute, Forschungstrasse 111, CH-5232 Villigen, Switzerland
\textsuperscript{2} The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong Special Administrative Region of China, People’s Republic of China
\textsuperscript{3} Department of Quantum Matter Physics, University of Geneva, 24 Quai Ernest-Ansermet, CH-1211 Geneva, Switzerland
\textsuperscript{4} Institute of Physics, Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland
\textsuperscript{5} Department of Physics, Eidgenössische Technische Hochschule Zurich (ETHZ), CH-8093 Zurich, Switzerland
\textsuperscript{6} Quantum Center, Eidgenössische Technische Hochschule Zurich (ETHZ), CH-8093 Zurich, Switzerland
\textsuperscript{7} Equal contribution.

\textsuperscript{*} Author to whom any correspondence should be addressed.

E-mail: sandy.ekahana@psi.ch

**Abstract**

There is a growing recognition that electronic band structure is a local property of materials and devices, and there is steep growth in capabilities to collect the relevant data. New photon sources, from small-laboratory-based lasers to free electron lasers, together with focusing beam optics and advanced electron spectrometers, are beginning to enable angle-resolved photoemission spectroscopy (ARPES) in scanning mode with a spatial resolution of near to and below microns, two- to three orders of magnitude smaller than what has been typical for ARPES hitherto. The results are vast data sets inhabiting a five-dimensional subspace of the ten-dimensional space spanned by two scanning dimensions of real space, three of reciprocal space, three of spin-space, time, and energy. In this work, we demonstrate that recent developments in representational learning (self-supervised learning) combined with \(k\)-means clustering can help automate the labeling and spatial mapping of dispersion cuts, thus saving precious time relative to manual analysis, albeit with low performance. Finally, we introduce a few-shot learning (\(k\)-nearest neighbor) in representational space where we selectively choose one (\(k = 1\)) image reference for each known label and subsequently label the rest of the data with respect to the nearest reference image. This last approach demonstrates the strength of self-supervised learning to automate image analysis in ARPES in particular and can be generalized to any scientific image analysis.

1. Introduction

Angle-resolved photoemission spectroscopy (ARPES) is a powerful tool to visualize the electronic band structure, which has been used mostly for condensed matter physics. Complexities in ARPES measurements and data analysis start to manifest themselves as measurements can now be performed in multiple dimensions almost simultaneously, demanding the development of automation and assistive tools to help the experiment. In short, ARPES evolved stepwise from one-dimension (1D) (photoelectron intensity (\(I\) vs. energy) available in x-ray photoelectron spectroscopy [1], to 2D where small 1D detectors are placed in an array creating a 2D analyzer slit for angle-resolved measurements (\(I\) vs. angle (\(\phi\), energy axes) (figure 1(c)) [2–6] with a typical resolution of \(\sim 0.1^\circ\), to 3D (1 vs. angle 1 (\(\theta\), angle 2 (\(\phi\), energy axes) (figure 1(d)) by either moving the 2D analyzer slit along the perpendicular angle or by rotating the sample itself [7–10]. This allows us to explore momentum space in two dimensions, i.e. (\(k_x, k_y, E\)) axis while the remaining perpendicular momentum \(k_z\) can be explored by varying the incoming photon energy (\(h\nu\)), thus completing the picture of the band structure in 3D solids [11, 12] where Fermi surface tomography can also be performed [13]. The other dimension like time (\(t\)) can be investigated [14, 15] depending upon the stimulus.
The spatial scan size, increasing the analysis time, \( \theta, \phi \) vs. angle (0 to 180) to start the experiment while the spin information, \( x, y \), is collected, i.e. (Intensity vs. \( x, y \), \( \theta \)) to create a 3D Fermi map. (e) The advent of a small beam spot allows a real-space scan creating 4D data where different band dispersion (Intensity vs. \( E_k, \theta \)) from different positions \( (x, y) \) are collected.

Figure 1. Angle Resolved Photo Emission Spectroscopy (ARPES) technique evolution. (a) ARPES is based on the photo-electric effect, where a single electron is ejected from a sample after one photon illuminates the sample. (b) The photoelectron kinetic energy can be detected with a 1D detector (Intensity vs. \( E_k \)). (c) Further development of the detector shows a 2D detector where electrons from different angle positions can be collected simultaneously (Intensity vs. \( E_k, \theta, \phi \)). (d) This 2D detector can be rotated to collect different band dispersion from different angular positions (Intensity vs. \( E_k, \theta, \phi \)) to create a 3D Fermi map. (e) The angle 1 (e.g., for laser-based ARPES) or simply the spatial inhomogeneities are electronic and not chemical and therefore only visible in the band structure, as for example in [23, 24]. Therefore, it is still desirable to perform a proper 4D-data measurement where for each spatial position we still have ARPES dispersion with good statistics from which we map the differences spatially (creating a spatial domain), trace the differences in a fine step (especially through the domain), and ultimately perform the analysis automatically in a memory saving way without any human intervention.

Labeling each band structure observed (Intensity vs. angle (e.g., sample cleave or laser pulse) applied at \( t = 0 \) to start the experiment while the spin information \( s_x, s_y, s_z \) carried by the photo-emitted electrons can be probed by a spin detector, e.g., VLEED [16] or Mott detector [17], establishing both in-plane and out-plane components.
decay time of a clean sample surface or simply because the statistics are limited by the time available for the experiment. Plotting the integrated intensity (including intensity normalization relative to some standard feature visible in each band structure slice) can be a preliminary step to draw and usually is the first-to-go technique. Subsequently, one may find a unique region of interest that can define a representative band structure distinct from others. Yet, this depends on the case-specific judgment and can take precious time during the measurement. Therefore, it is ultimately favorable to have an automated procedure to help the labeling/clustering of different band structures so we can focus on the physics problem in an efficient, unbiased fashion.

Meanwhile, there have been attempts to implement techniques available in the machine learning field to scientific experiments like scanning tunneling [25–28] and atomic force microscopies [29, 30], where the majority entails supervised models that need manual labeling, to begin with. There is also similar work for various ARPES applications. For example, a deep-layer convolutional neural network (ConvNet) has been trained to denoise ARPES data [31]. Afterward, there have been efforts to determine how the band structure calculation is related to the ARPES data [32, 33], where the [33] provides an additional feature for obtaining the result even through noisy data (simulated noise). There has been also work in automation of spatial domain assignment with a smaller subset of data over a predetermined area, where subsequent position measurement is calculated with the Gaussian process to give the possible highest amount of information [34]. Our work here is in the same direction as [34], which is to automate domain assignment. However, our approach has a stark difference: we use representation learning from self-supervised models to represent our ARPES images.

In the context of computer vision, we need to distinguish the term ‘to classify’ and ‘to cluster’. The term classify usually refers to supervised labeling where there is a definite set of labels for these band structures. For example, we can have a clear pre-defined label of ‘with-gap’ and ‘no-gap’ band structure which each ARPES cut can be categorized as. Meanwhile, clustering refers to the same-ness, like-ness, or affinity to a set of standard images to which band structures are compared; different image references will create their different clusters. From the ARPES perspective, the final product of ‘to cluster’ and ‘to classify’ can be identical, which is a list of numbers indicating the group this band structure belongs to, yet the methods to produce the list are technically different.

Naturally, one may resort to the automated labeling technique called supervised learning, e.g. convolutional neural network (ConvNet) [35] in this case and come up with a trained neural network model that can classify the given band structure with a well-defined label. The ConvNet is a well-known and robust machine learning model to learn image representation by learning from examples. Despite this, the ConvNet model may lack generality in the low-resource setting when the model is trained with very limited data and we need to train the model from scratch. Meanwhile, the amount of data from each measurement may not be sufficient for a neural model to be trained properly, despite the many available data enhancement procedures [33].

2. Supervised learning in ARPES

We may argue that any well-known pre-trained deep neural network like ResNet50, which is pre-trained with ImageNet dataset, can be used in the model initialization, where subsequent training with our training dataset will be done; the processes are described in figure 2, and the details about the dataset used are in the supporting information section B (SI.B). In short, we run the evaluation using k-fold cross-validation, and the performance metrics are calculated by taking the average performance on all folds. In this approach, we set the result in a probabilistic manner for each label (total of four labels) on one image. The performance of this fine-tuned ResNet50 is evaluated by the metrics, such as accuracy, F1, precision, and recall, as shown in table 1; refer to appendix for the definition of each metric element. The approach clearly demonstrates some usefulness as the accuracy, and F1 is relatively high, given the sparsity of the training data. Thus, this positive result suggests leveraging pre-trained models for ARPES data in supervised learning scenarios. In figure 2, we display how all the data are subsequently predicted by fine tuning the ResNet50 model, where the group probability of each real space position is shown. We can see that the result of the ResNet50 prediction can be used as a guide to investigating different domains in the experimental context. It also shows that the ResNet50 trained with generic images can handle experimental data images with small fine-tuning. However, it should be restated that this approach lacks generality as the fine-tuned ResNet50 model can only be used for the Fe$_3$Sn$_2$ data; a new sample needs a new re-training which implies the need for a new set of labeled data, which is precisely what we aim to avoid. Following this, the retraining can be expensive computationally (involving GPU) and not something favorable to be performed during the ARPES measurement itself as it also costs valuable time.
Figure 2. Application of supervised learning to Fe$_3$Sn$_2$ ARPES 4D data. (a) The pre-trained ResNet50 model is re-trained with a training dataset (a fraction of total data) of labeled Fe$_3$Sn$_2$ band dispersion (four labels in total as output). (b) The re-trained ResNet50 model is used to predict the rest of the Fe$_3$Sn$_2$ ARPES 4D data with an output of probability for each group. We can see that group 2 dominates the data population. The domain picture from the maximum probability label is also shown.

Table 1. Results of fine tuning pre-trained ResNet50 on Fe$_3$Sn$_2$ 4D ARPES data.

| Method  | Fold | Accuracy | F1   | Recall | Precision |
|---------|------|----------|------|--------|-----------|
| ResNet50| 1    | 90.24    | 73.99| 67.65  | 89.13     |
|         | 2    | 93.06    | 81.31| 75.43  | 94.20     |
|         | 3    | 90.67    | 77.38| 72.74  | 86.27     |
|         | 4    | 89.15    | 73.95| 71.24  | 82.86     |
|         | 5    | 88.89    | 77.68| 73.33  | 83.45     |
| AVG     |      | 90.40    | 76.86| 72.08  | 87.18     |

3. Unsupervised clustering in ARPES

Another trivial attempt in the automation can be made by an unsupervised clustering method, such as $k$-means [36], where the data can be clustered according to their affinity to each other; usually, the Euclidean distance of the feature dimension is used to define the affinity. The typical data processing done before applying this procedure is to flatten the picture, i.e. convert the 2D array of the image into a 1D array and perform the $k$-means clustering in this ‘hyper-dimensional’ array. However, the ‘curse of dimensionality’, as coined by Bellman [37], might play a role where this many-dimensional vector effectively hides the feature that needs to be captured. For this issue, one may do the common dimension reduction technique such as principal component analysis [38] or a $t$-distributed stochastic neighbor embedding ($t$-SNE) [39] to reduce the number of dimensions and subsequently perform the $k$-means clustering on this reduced dimension space. This approach may help solve the curse of dimension problem, but table 2 shows that the performance metric is still comparable when the image data is simply flattened and clustered (method = ‘none’ in table 2). In any case, some kind of ‘flattening’ is still done on the picture where the object of interest is locked into some position in the array. This will greatly affect the result as the feature of interest needs to be on the same array pointer, and some data preprocessing needs to be done to keep the features in place before flattening.

4. Self-supervised learning in ARPES

Recently, pre-trained self-supervised models are utilized to generate high-dimensional representation of data, where instead of comparing the input data directly, the model will map the input data onto a representational space on which further training can be done, or simply use them for clustering [40]; the input data can be speech, text, or images (refer to supporting information for details). In short, these models learn the feature from the input data in a self-supervised way, which means the training objective is generated without using annotated human labels but using pretext tasks. In the context of natural science experiments, there are already attempts to apply some form of self-supervised learning, for example in the training of self-supervised model built for single cell image analysis [41], training of self-supervised models to denoise tomography data (Noise2inverse) [42, 43], analysis of electrocardiography database with a general
Table 2. Results of $k$-means clustering using pre-trained model and none as bare-image, with and without dimension reduction by t-SNE. Largest values are in bold.

| Method | Dimension reduction | Accuracy | F1 | Recall | Precision |
|--------|---------------------|----------|----|--------|-----------|
| DINO   | No                  | 50.79    | 46.59 | 61.73  | 49.95     |
| SwAV   |                     | 61.25    | 59.97 | 68.74  | 59.91     |
| MoCo   | No                  | 53.90    | 50.28 | 65.99  | 51.96     |
| BYOL   |                     | 57.05    | 57.58 | **74.40** | 58.24     |
| None   |                     | 53.42    | 54.61 | 72.12  | 59.14     |
| DINO   | Yes (t-SNE 2-d)     | 43.80    | 40.61 | 62.23  | 45.55     |
| SwAV   |                     | 49.28    | 40.60 | 60.59  | 47.09     |
| MoCo   | Yes (t-SNE 2-d)     | 50.10    | 41.92 | 59.34  | 43.81     |
| BYOL   |                     | 46.83    | 43.56 | 62.50  | 47.86     |
| None   |                     | **58.97** | **48.97** | **67.48** | **48.72** |

Figure 3. Clustering pipeline. The raw ARPES images are to be clustered with the $k$-means algorithm. Path 1 follows the procedure where the raw ARPES images are directly flattened and used for the $k$-means algorithm, with and without dimension reduction (t-SNE onto 2-dimension). Path 2 follows the procedure where additional ARPES image conversion into representational space (rep-space) is done prior to $k$-means clustering, with and without further dimension reduction of rep-space. The final results of all procedures give us the collection of labels for each input image.

Purpose self-supervised model [44], quantifying hidden features in single-molecule charge transport data with transfer learning [45], etc. Meanwhile, the field of computer science has already introduced several general-purpose self-supervised models for computer vision, such as MoCo [46], SimCLR [47], BYOL [48], SwAV [49], and DINO [50], which ARPES can take advantage of through transfer learning, i.e. utilize a pre-trained model that was trained using self-supervision fashion on large image datasets (e.g., ImageNet [51]). The summary of the procedure is described in the supporting information and the clustering pipeline is shown in figure 3. Table 2 shows the result of $k$-means clustering performance metrics done on the representation space of DINO, SwAV, MoCo, BYOL, with and without t-SNE dimension reduction. We can see that the performance is worse than the ResNet50 supervised performance (table 1) and unfortunately it is even comparable to not using a pre-trained self-supervised model (method = ‘none’) on the image; dimension reduction also offers no significant help in this approach. This result may imply that the features of ARPES images are not captured well in the knowledge extracted from the ImageNet database; this invites the need for a collective ARPES database for training purposes and also a novel objective task of self-supervised models for ARPES specifically or for any scientific method which relies heavily on images. It can also mean that ARPES images from the Fe$_3$Sn$_2$ dataset are not well separated in the representation space; the representation learning space on images may be well spread, yet they do not make well-separated islands on which $k$-means clustering can be of any use.

As a current solution, we introduce a $k$-nearest neighbor (kNN) procedure (the pipeline is shown in figure 4), where instead of direct clustering on the representation space, we take $k$-numbers of images as references to which the rest of the images can be compared. The references here are the images with a known label set by the expert. The input images and the references are first converted into the representation space.
by passing to a pre-trained self-supervised model accordingly and the Euclidean distances of the converted images are measured with respect to the converted references. The label is then determined by the closest reference. In our case of Fe$_3$Sn$_2$, we have four labels and thus $k$-numbers of images on each label. In the case of $k = 1$, we can see that this approach is equivalent to generating $k$-means clustering prediction but with a pre-defined centroid (four pre-defined centroids for the case of Fe$_3$Sn$_2$). This problem can be a quick solution in the case when the images in the representation learning space are not well separated yet they are well spread. As we increase $k$, we expect the model to be more able to discriminate the given input. The pipeline of this process is drawn in figure 4.

Table 3 is the summary of the kNN performances. We can see that as we set $k = 20$ their performances are roughly similar to each other with ranking as follows: DINO, none, BYOL, MoCo, SwAV. This result is expected because as more known labels are given, we are approaching more towards a supervised method like ConvNet above. This result also tells us that different representational techniques capture different features of ARPES data differently as their performances vary between techniques. Importantly, we can see that the DINO technique captures ARPES features better than the other techniques. Reducing the $k$ down to the single reference case $k = 1$ also reveals the strength of the DINO model [50] (we average 50 different

Table 3. Results of the kNN few-shot experiment using pre-trained representation learning. Largest values are in bold.

| Method | K  | Accuracy | F1  | Recall | Precision |
|--------|----|----------|-----|--------|-----------|
| DINO   | 1  | 80.36    | 70.66 | 80.56  | 70.84     |
|        | 5  | 89.16    | 82.09 | 90.62  | 78.73     |
|        | 10 | 91.23    | 85.14 | 93.26  | 81.14     |
|        | 20 | 92.94    | 87.65 | 95.16  | 83.43     |
| SwAV   | 1  | 71.37    | 61.59 | 73.13  | 64.49     |
|        | 5  | 82.12    | 74.35 | 85.35  | 72.11     |
|        | 10 | 84.62    | 77.56 | 88.62  | 74.13     |
|        | 20 | 87.36    | 80.71 | 91.34  | 76.56     |
| MoCo   | 1  | 69.74    | 59.25 | 71.19  | 61.36     |
|        | 5  | 82.17    | 72.74 | 84.37  | 69.21     |
|        | 10 | 85.89    | 77.27 | 87.98  | 72.84     |
|        | 20 | 88.32    | 80.51 | 90.48  | 75.71     |
| BYOL   | 1  | 76.61    | 65.82 | 75.36  | 68.22     |
|        | 5  | 84.68    | 76.75 | 86.72  | 73.96     |
|        | 10 | 86.89    | 79.74 | 89.67  | 75.96     |
|        | 20 | 88.71    | 82.21 | 92.07  | 77.86     |
| None   | 1  | 76.41    | 66.81 | 77.17  | 68.78     |
|        | 5  | 83.94    | 76.39 | 88.21  | 72.63     |
|        | 10 | 87.00    | 80.03 | 91.38  | 75.35     |
|        | 20 | 90.06    | 83.90 | 94.02  | 78.90     |
single references for each technique) as the performance still compares favorably to the other techniques. We display the attention map of DINO in the SI for further visualization on how it works.

From the kNN results, we propose the following pipeline to finally solve our problem of automated domain labeling in pseudocode 1. As the spatial scan begins, we take the first image as our reference \( k = 1 \) case. Afterward, the subsequent measured images’ affinity to the reference is calculated (in representation learning space). For each measured image, the affinity is considered if it has exceeded a certain threshold. The threshold itself is an arbitrary value that will be determined as the experiment runs (the expert may intervene during the experiment to decide the threshold value). At some point, the Euclidean distance of an image might be relatively further than the others. In this situation, we (as the expert during the measurement) may decide if the new image is regarded as a new label or not, and thus we may determine if a new label is needed. Afterward, the experiment again proceeds with the updated number of labels. Finally, the experiment loop will end with a complete set of ARPES data set and their labels as measured with respect to the decided references. The method described here is still not fully automatic. Nonetheless, it is a semi-supervised way of automation and is already a significant improvement with the help of the learned representation from self-supervised models.

--- pseudocode 1. kNN pipeline proposal ---

%% pseudocode for kNN pipeline %%

Require: \( M \) as the list of ARPES matrices
Require: \( L \) as the number of labels
Require: \( r \) as the list of ARPES reference images, where \( r_i \) is the image for label \( i \)
Require: \( \Theta \) as the pretrained model

for all \( m \) in \( M \):
    \( x = \Theta(m) \) # get the representation of an image \( m \)
    \( \text{ref}_{\text{ref}} = \Theta(r) \)
    \( \text{closest}_\text{label} = \arg\min_i \text{to } L (\text{euclidean}_\text{distance}(x, \text{ref}_{\text{ref}})) \)
    if \( \text{closest}_\text{label} > \text{threshold} \):
        prompt_user
End for

--- pseudocode 1. kNN pipeline proposal ---

5. Conclusion

We demonstrate that ARPES, which relies heavily on pictorial data analysis, can take advantage of the recent development of representational learning in the computer vision field for help in automation. In this work, we demonstrate the transfer learning application of pre-trained self-supervised models that are applied to ARPES images in supervised and unsupervised manners. We apply the KNN method, where the affinities of the ARPES images are measured with respect to reference images, and we show that using only a single reference image can achieve acceptable performance. The KNN proposal presented here is not limited to ARPES image analysis and can be used for other experimental techniques that yield images and sparsely labeled data. Our work urgently invites the creation of an expanded ARPES image database on which machine-learning models can be trained. This will entail further interdisciplinary research collaborations between ARPES and computer vision to find more suitable representational learning methods for ARPES. We anticipate that similar paradigms will advance the analysis of other measurements producing high dimensional data sets.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

We thank Benjamin Bejar Haro from Laboratory for Simulation and Modelling, Paul Scherrer Institute, for the insightful discussion on the k-shot model.
We thank Yao Mengyu for his participation in the ARPES measurement.
We thank Yihao Wang and Yimin Xiong for providing us with the sample.
We thank Felix Baumberger for providing access to the micro-focused laser ARPES setup.
We thank Matthew Watson from i05 Diamond Light Source for the valuable discussion in spatially resolved ARPES.
We thank Vincent Tatan for the early discussion on the ConvNet technique.

S A E and G A acknowledge the support from the European Research Council HERO Synergy Grant SYG-18 810451, NCCR-MARVEL funded by the Swiss National Science Foundation, the European Union’s Horizon 2020 research and innovation program under the Marie Skłodowska-Curie Grant Agreement No. 701647.

Ethical statement

The study does not include human subjects, human data or tissue, or animals.

Conflict of interest

The authors declare no competing interests.

Appendix. Definitions

| Truth condition | Positive (P) = TP + FN | Predicted Positive (PP) = TP + FP | Predicted Negative (PN) = FN + TN |
|-----------------|------------------------|-----------------------------------|-----------------------------------|
|                 | True positive (TP)     | False negative (FN)               | False positive (FP)               |
|                 | True negative (TN)     |                                    | True negative (TN)               |

Accuracy (ACC)

\[ \text{ACC} = \frac{TP + TN}{P + N}. \]

F1

\[ \text{F1} = \frac{2TP}{2TP + FP + FN} = \frac{2 \times \text{PRE} \times \text{REC}}{\text{PRE} + \text{REC}}. \]

Recall (REC) or true positive rate (TPR) or sensitivity (SEN)

\[ \text{REC} = \frac{TP}{P} = \frac{TP}{TP + FN}. \]

Precision (PRE) or positive predictive value (PPV)

\[ \text{PRE} = \frac{TP}{PP} = \frac{TP}{TP + FP}. \]

From the definition above, imbalanced data may create stark differences between REC and PRE as in our \( \text{Fe}_3\text{Sn}_2 \) example, where this value is reflected in F1. In this case, F1 is a good indicator for the goodness of the model as it ‘averages’ REC and PRE.

ORCID iDs

Sandy Adhitia Ekahana https://orcid.org/0000-0002-1693-8082
Anna Tamai https://orcid.org/0000-0001-5239-6826

References

[1] Nordling C, Sokolowski E and Siegbahn K 1957 Precision method for obtaining absolute values of atomic binding energies Phys. Rev. 105 1676–7
[2] King D M et al 1993 Fermi surface and electronic structure of \text{Nd}_2-x\text{Ce}_x\text{CuO}_4 \_d Phys. Rev. Lett. 70 3159–62
[3] Park C H, Shen Z-X, Loeser A G, Dessau D S, Mandrus D G, Migliorini A, Sarrao J and Fisk Z 1995 Direct observation of a narrow band near the gap edge of \text{FeSi} Phys. Rev. B 52 16981–4
[4] Shen Z X and Schrieffer J R 1997 Momentum, temperature, and doping dependence of photoemission lineshape and implications for the nature of the pairing potential in high-\( T_c \) superconducting materials Phys. Rev. Lett. 78 1771–4
[5] Damascelli A et al 2000 Fermi surface, surface states, and surface reconstruction in \text{Sr}_2\text{RuO}_4 Phys. Rev. Lett. 85 5194–7
[6] Cuk T et al 2004 Coupling of the \( \text{B}_1 \)g phonon to the antinodal electronic states of \text{Bi}_2\text{Sr}_2\text{Ca}_{0.92}\text{Y}_{0.08}\text{Cu}_2\text{O}_8+\delta Phys. Rev. Lett. 93 117003
[7] Liu Z K et al 2014 A stable three-dimensional topological Dirac semimetal \text{Cd}_{3}\text{As}_2 Nat. Mater. 13 677–81
