

# CHAPTER 1

## Defects and CDW interplay in Hf-doped ZrTe<sub>3</sub> across temperature

### 1.1 Motivation

In the previous chapter, we investigated the structural properties of defects in ZrTe<sub>3</sub>. We now explore the evolution of the **charge density wave (CDW)** with temperature. In particular, we are interested in the interplay between defects and the CDW below and above  $T_{\text{CDW}}$ .

To analyze this, we need to compute the **cross-correlation** between the defect signal and the CDW signal. Extracting the CDW signal is straightforward and can be achieved by filtering the **FFT CDW signal**. However, extracting the defect signal is more challenging since isolating an **aperiodic signal** is not trivial.

To address this issue, we developed **three distinct methods**, which we will discuss in detail below. But first, we examine the evolution of the CDW state across different temperatures.

### 1.2 Temperature Evolution of CDW in ZrTe<sub>3</sub>

The charge density wave (CDW) in ZrTe<sub>3</sub> is unidirectional, propagating along the *a*-axis with a wave vector of  $0.07a^*$ . In our measurements of Hf-doped ZrTe<sub>3</sub>, we observe that the CDW remains uniform below the transition temperature  $T_{\text{CDW}}=63\text{K}$ . However, above the transition temperature the CDW state is still evident but appears most clearly near defects. (see Figure 1.1) and translate to a cloudy area around the main CDW signal (see figure1.2) .

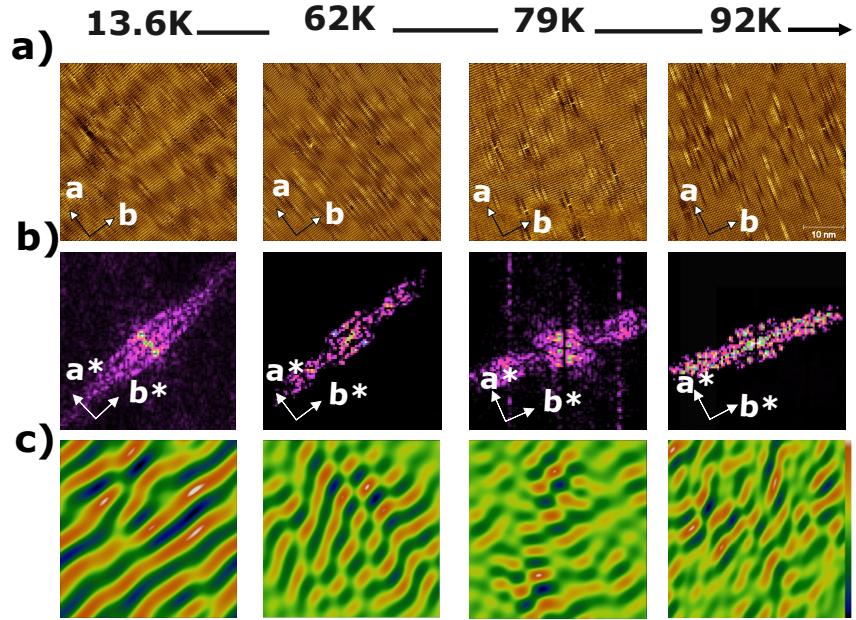


Figure 1.1: Temperature evolution of the CDW. (a) Topographical image taken at  $-50$  mV,  $I = 50$  pA, with a scan size of  $45 \times 45$  nm $^2$ . (b) FFT of the corresponding image. (c) Fourier-filtered CDW signal extracted from (b).

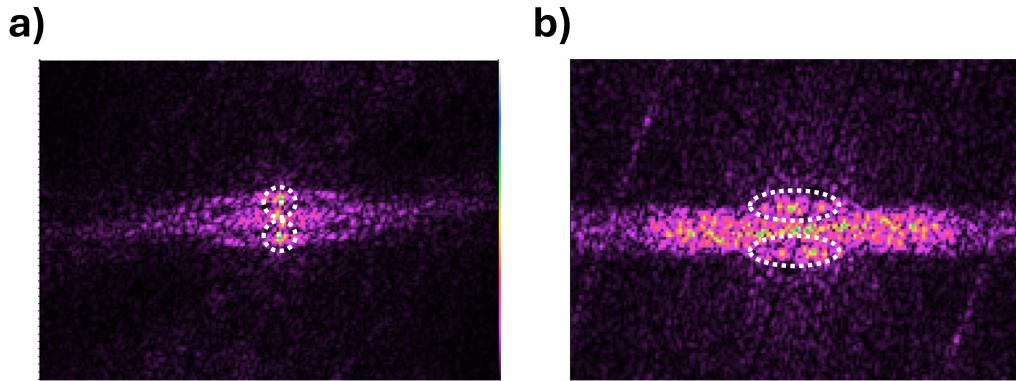


Figure 1.2: Fast Fourier Transform (FFT) of the CDW below and above the transition temperature. (a) Zoomed-in FFT from Figure 1.1(a) at **13.6 K**, showing sharp CDW peaks. (b) Zoomed-in FFT from Figure 1.1(a) at **92.6 K**, showing a diffuse CDW signal.

## 1.3 Defect Signal Extraction

At higher temperatures, the **CDW** appears to be localized near defects, suggesting that defects may act as **nucleation** or **pinning sites**.

Here, we provide a quantitative approach to understanding the relationship between the **CDW state** and the **defects**.

The identification and extraction of the defect signal is a crucial step in performing the **cross-correlation** calculation. Extracting the CDW signal is relatively straightforward, as it can be isolated by filtering the **CDW component** from the **Fast Fourier Transform (FFT)** (see Figure 1.2). However, extracting the defect signal is more challenging because defects may appear slightly different depending on **tip conditions** and **bias voltage**.

To address this challenge, we developed three techniques:

1. FFT Signal Extraction
2. Template Matching Method
3. Defect Classification Using Machine Learning

These techniques will be detailed in the following sections. For the purpose of this study, we primarily used template matching, as it provided similar results to the FFT method. The machine learning approach yielded very promising results; however, it requires an extensive labeling process, which is highly time-consuming.

### 1.3.1 FFT Signal Extraction

This method primarily involves filtering the defect signal in the FFT. When inspecting the FFT of raw topographies, we observe a diffuse cloud extending along the  $b^*$ -direction, which is associated with the defect signal. However, filtering only this region may result in the loss of some defect-related information, causing defects to appear more elongated than they actually are, making it difficult to effectively mask them (see Figure 1.3). To overcome this, we apply a rectangular filter at the low wave-vector region while ensuring that the CDW region remains unfiltered. This results in an image that is free of both atomic and CDW signals. To further reduce noise, Gaussian smoothing can be applied—for instance, in Figures 1.4(b-c), a 3-pixel Gaussian filter is used.

Following this, we proceed by separately masking the bright and dark defects. The extracted masks can then be processed using a custom MATLAB script, which identifies the maximum intensity peaks, isolates the defect signal from the noise, and calculates the center of each defect. Figures 1.4(e-f) illustrate the centers of both bright and dark defects overlaid on the raw topography.

While this method is effective, one limitation is that the filtering parameters must be manually adjusted for different images, potentially introducing minor bias. Additionally, when image quality is low and affected by tip changes, which introduce striped noise unique to STM images, noise signals may be misinterpreted as defect signals. Furthermore, slight inaccuracies in center calculation may arise due to these challenges, as the defect centers should ideally align with Te vacancies for dark defects and Zr subsurface vacancies for bright defects.

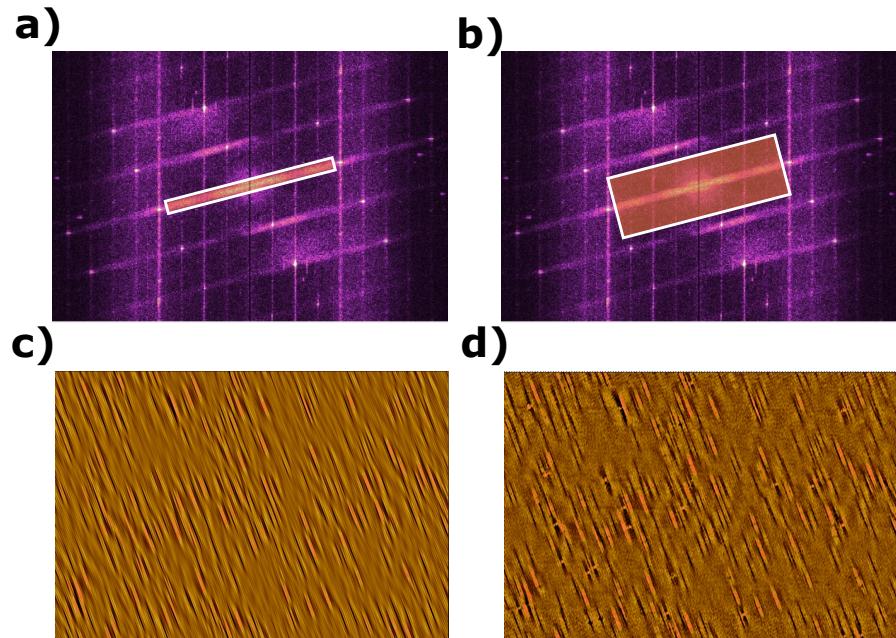


Figure 1.3: FFT defect filtering comparison. (a) FFT of the topography shown in Figure 1.4(a), with a partial defect signal region highlighted by a rectangle. (b) FFT of the topography shown in Figure 1.4(a), with the full defect signal region highlighted by a larger rectangle. (c)-(d) Fourier-filtered and masked representations of the bright defect from the FFTs in (a) and (b), respectively. This figure highlights the importance of carefully applying Fourier filtering to the defect signal to fully capture its entire contribution.

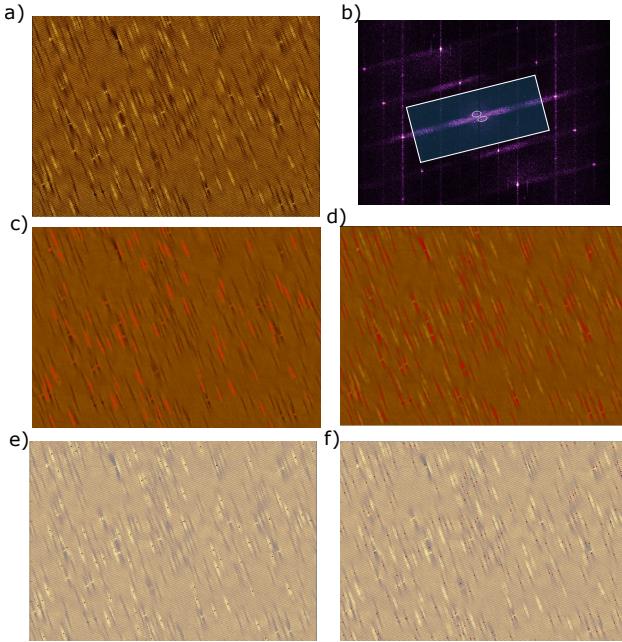


Figure 1.4: FFT defect signal extraction method. (a) High-resolution topography acquired at  $V = -50$  mV,  $I = 50$  pA, with a scan size of  $60 \times 92$  nm $^2$ . (b) FFT of the image shown in (a), with the defect signal region highlighted by a large rectangle while removing the inside center (CDW signals). (c)-(d) Fourier filtered topography, Masked bright and dark defects, respectively. (e)-(f) Post-processing of the masked defects to extract their centers and reduce noise with MATLAB.

noting that this method needs to be adapted based on image quality

### 1.3.2 Template Matching

We have used a second method to extract defect locations using the **template matching method**, following several steps.

#### Step one

First, we rotate the image until the defects align parallel to the **Y-axis**. Next, we set the intensity to zero and normalize it from 0 to 1 using arithmetic operations in **Gwyddion**. Afterward, we apply a **Gaussian filter** with a radius of 9 pixels (or until the atomic signal is no longer visible) to ensure consistency across all images in the study. This can be achieved either by directly applying a Gaussian filter or by selectively unfiltering the atomic signal.

Finally, it is essential to subtract the **CDW signal** from the image to isolate the defect signal effectively.

### Step two

In the next step, we crop a single defect and save both the defect and the entire image as text files within the pattern-matching code environment. The code then computes the **cross-correlation** between the single defect and the entire image.

The **cross-correlation** is a mathematical technique used to measure the similarity between two signals. It quantifies how closely one signal resembles another when shifted in space or time. In this context, we use cross-correlation to compare the **template**, which represents a single defect, with the rest of the defects within the topography. A detailed description of the function 1.1 will be provided later in this chapter. The **cross-correlation function** multiplies and normalizes the outcome between the matrix representing the defect and the corresponding portion of the image matrix. Subsequently, the code slides the defect template across the entire image to identify and match all defects resembling the initially cropped defect.

Additionally, the code crops the cross-correlation map to the size of the original image and generates a binary mask of all defects matching the initial defect. Before computing the mask, a threshold for high correlation must be set. In this study, the threshold used ranged between 0.5 and 0.65.

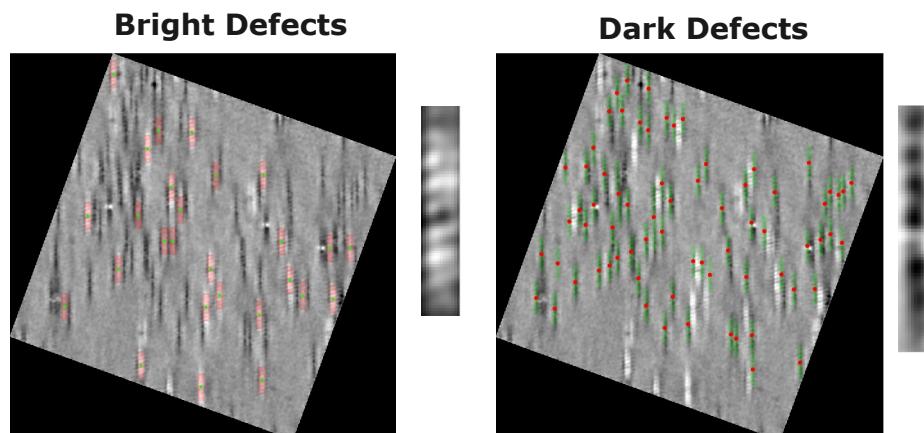


Figure 1.5: Template matching method. Rotated topography with corresponding template of bright and dark defect to use to compute normalized cross correlation to identify bright and dark defects location

Step one is crucial, as it ensures an image free of atomic features. The periodicities of the atoms within the image will correlate with those of the template, ultimately leading to incorrect defect pinpointing in the final results.

### 1.3.3 Defect Classification Using Machine Learning

In order to extract the defect signal and study the interplay between defects and the charge density wave (CDW), we used two main techniques: **Template Matching** and the **Fast Fourier Transform (FFT) method**, as described previously.

In addition to these methods, I explored a **machine learning approach** using **YOLOv8**, a deep learning algorithm developed by **Ultralytics**, which is primarily used for object detection and classification. YOLOv8 has been widely applied in tasks such as facial recognition and general object detection in videos and images[1][2][3]. It has also been utilized in biological applications to identify and distinguish various cells and viruses[4][5]. However, this type of deep learning tool has not been widely applied to defect classification in **two-dimensional (2D) materials**.

In this section, I will explain the steps involved in generating training images, labeling them, and training the neural network to classify **bright and dark defects**. Even though the results were promising, they have not yet been integrated into cross-correlation calculations. Nevertheless, this approach has been extremely useful in verifying our defect density calculations.

As illustrated in Figure 1.6[6], object detection can differentiate between different categories, such as cats and dogs, even though individual cats may look different from one another, and dogs may vary slightly in appearance. This analogy can be applied to our project: while bright and dark defects exhibit distinct characteristics, variations in tip conditions and different defect interplay scenarios can cause a dark defect to appear slightly different from another dark defect, and similarly for bright defects.

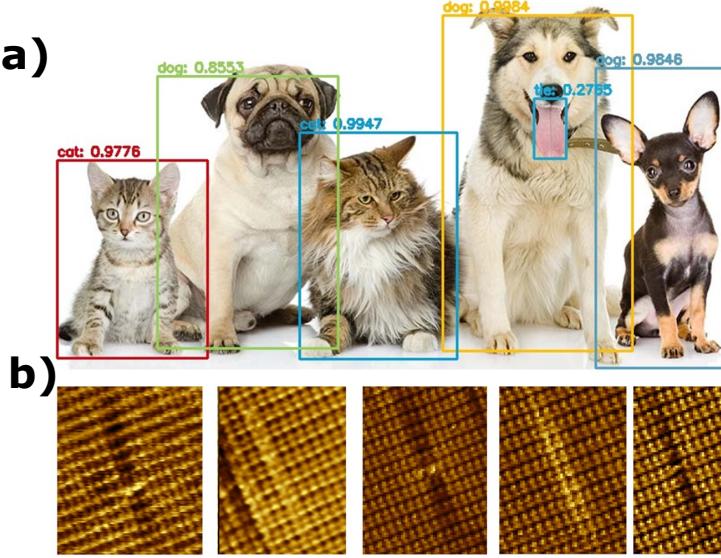


Figure 1.6: Analogy between pet and defect classification. (a) Example of object detection distinguishing between different categories, such as cats and dogs[6]. (b) defects appear differently depending on experimental conditions.

## Quick Introduction to Machine Learning for Image Analysis

For this brief introduction to Convolutional neural networks, I will use the simplest example of a neural network including only a single node.

In a single node of a neural network, the prediction represents solving only a first-degree equation, as shown below:

$$Y = W \cdot X + b$$

where  $X$  is the input in an array shape,  $Y$  is the output in an array shape as well, and  $W$  and  $b$  are the weight and bias, respectively, which are the trainable parameters. See the diagram below:

This is the simplest way to understand a single node in a neural network operation. To scale the neural network, we can add multiple nodes with multiple weights and biases. The weights initially start randomly, and after many iterations, with the right  $W$  and  $b$ , the neural network should be able to predict  $Y$  based on the input  $X$ .

In our case, we aim to train a convolutional neural network (CNN), where the input represents an image in matrix form. This matrix is processed through a series of convolutional operations, reducing its dimensionality while extracting relevant features. The final output,  $Y$ , represents the predicted result. In the case of segmentation tasks,

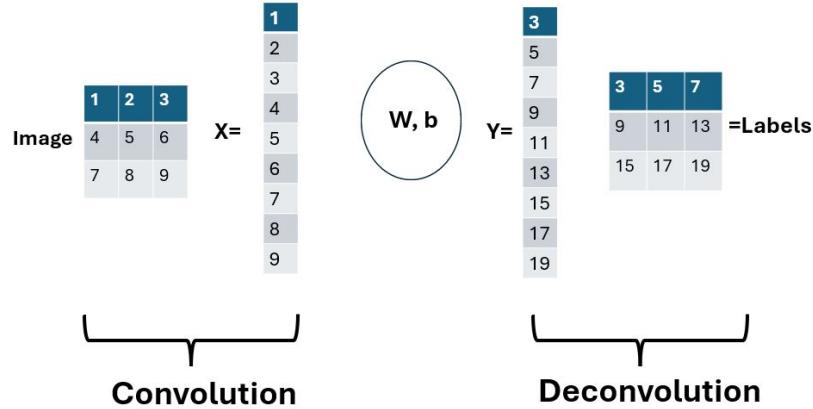


Figure 1.7: Illustration of a single-node and simplified Convolutional neural network operation.

the output undergoes a deconvolution step to reconstruct the desired segmentation masks.

To measure the progress and accuracy of a neural network is to plot the epoch (iteration) vs. the loss function, where the loss function is given by:

$$Loss = \frac{1}{N} \sum_{n=0}^N (Y_i + Y_0)$$

where  $Y_0$  is the model output, and  $Y_i$  are the predictions. A good loss function should be close to zero.

## Model Training Steps

Training the model requires four main steps:

1. **Manual data augmentation**
2. **Defect labeling**
3. **Partitioning the dataset into training, validation, and test batches**
4. **Data augmentation using Roboflow**

### Manual Data Augmentation

The purpose of manual data augmentation is to expand the training set and mimic how images appear at various temperatures and under different tip conditions. For this methodology, I used data captured at 91K and cropped them into 16 equal-sized

segments of  $22.5 \times 22.5$  nm. These segments were then resampled to 640 pixels to optimize neural network performance.

The data augmentation process 1.8(a-d) involved three main operations:

- Lowering the atomic signal
- Reducing the defect signal
- Subtracting the atomic signal

### Defect Labeling

For the second step, defect labeling, I utilized **Roboflow**[7] (see Figure 1.8(e-h)), a labeling tool that provides features for annotating images and performing data augmentation. Roboflow offers functionalities such as adding noise, rotating, and mirroring images, which help increase the size of the training set and enhance neural network accuracy.

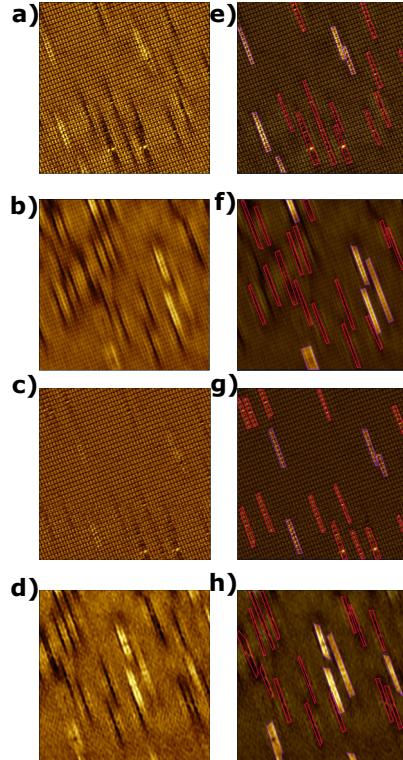


Figure 1.8: Defect labeling[7]. (a) Raw image. (b) Fourier-filtered image of (a) with low defect signal. (c) Fourier-filtered image of (a) with no defect signal. (d) Fourier-filtered image of (a) with no atomic signal. (e-h) Labeled corresponding images distinguishing between bright and dark defects.

## Partitioning the Dataset

For the third step, partitioning the dataset, we need to divide our data into training, validation, and test sets. This can be done manually by adding lines of code or using **Roboflow** just before generating the final dataset. I found it more beneficial to use Roboflow for dataset splitting because it offers a **health check** option, which evaluates whether the dataset partitioning is optimal for training.

In our case, the data were partitioned as follows:

- **Training set:** 29 samples
- **Validation set:** 22 samples
- **Test set:** 12 samples

## Data Augmentation

For data augmentation, we also had the option of using **Roboflow**. However, with the free version, the number of augmentations was limited. For future studies, we may explore data augmentation using Python packages to overcome these limitations.

For this study, the training data was augmented from 64 to 121 images. The augmentation process incorporated:

- Noise addition (up to 1.84% pixels)
- Blur (up to 3.8 pixels)
- Brightness adjustments
- Cropping
- Rotation (between  $-15^\circ$  and  $+15^\circ$ )

Below, we present the plot of the loss vs. epoch (see figure 1.9 (a,b)) for the training of the bounding boxes, as well as the training of the segmentation between dark and bright defects.

In addition, we show the confusion matrix (see figure 1.9 (c)), where **EDD** represents Extended Dark Defects, and **EBD** represents Extended Bright Defects. We observe that the background contributes significantly to the segmentation, indicating that the neural network struggles to clearly distinguish between dark defects and the background. This can be explained by the dataset we created, which includes a substantial number of images with low defect signals.

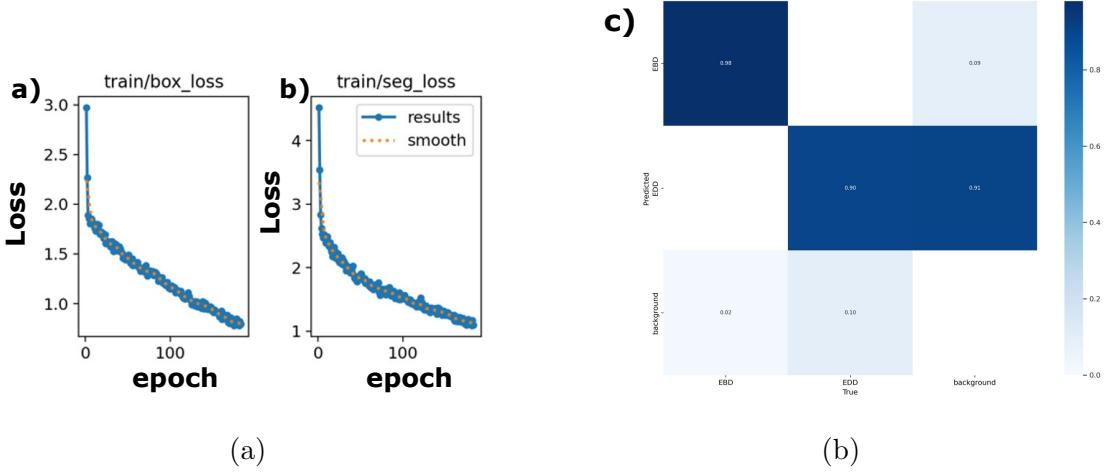


Figure 1.9: Convolutional neural network model performance parameters.(a and b) Loss vs. epoch during training. (a) Training loss for bounding boxes. (b) Training loss for segmentation. The loss decreases over epochs, indicating model convergence. (c) Confusion matrix for defect classification. The matrix represents the model's performance in distinguishing between Bright Defects (EBD), Dark Defects (EDD), and the background. Higher values along the diagonal indicate accurate classification, while off-diagonal values represent misclassifications.

The model performs well with high-resolution data, as shown in Figure 1.11(a-b). However, for low-resolution data, such as in subfigure (e), the model struggles to identify the full shape of dark defects.

This performance can be further improved by implementing **cross-validation**. The objective is to enable the model to randomly select data for training and validation, improving its generalization. Another approach would involve retraining the model using larger-sized images acquired at different temperatures. To achieve this, a larger number of images would need to be relabeled before training.

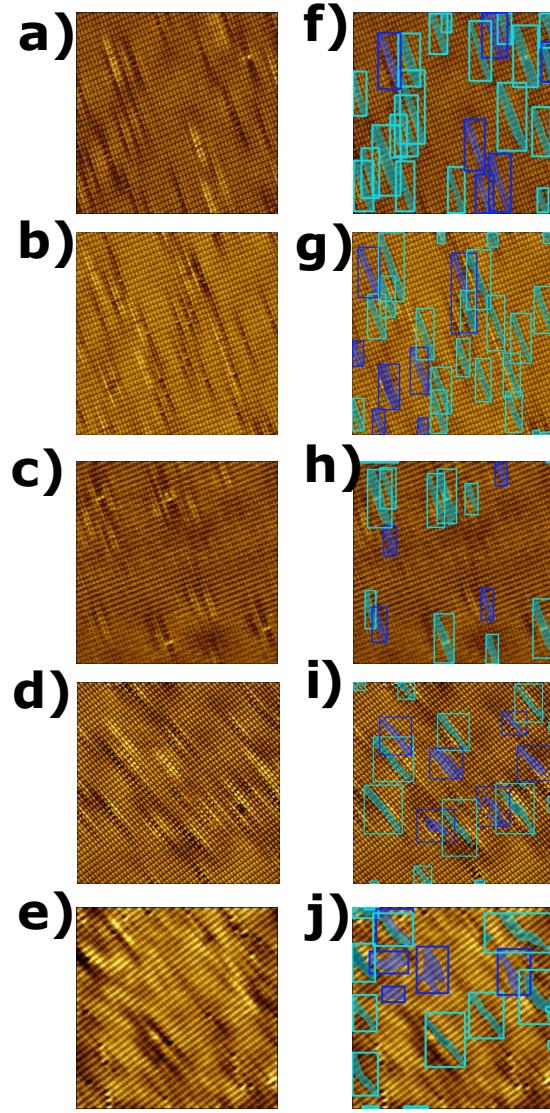


Figure 1.10: YOLOv8 model predictions. (a-e)  $45 \times 45 \text{ nm}^2$  images taken at different temperatures: 90K for (a) and (b), followed by 80K, 60K, and 13K, respectively, under the same scanning conditions ( $V = -50 \text{ mV}$ ,  $I = 50 \text{ pA}$ ). (f-j) Corresponding label predictions.

## 1.4 Defect Density Calculations and Defect Formation Energy

To calculate the defect density, we rely on the defect signal extraction methods discussed previously (see Figure 1.11). In  $45 \times 45 \text{ nm}^2$  images, we detect approximately

	<b>Fig(a)</b>	<b>Fig(b)</b>	<b>Fig(c)</b>	<b>Fig(d)</b>	<b>Fig(e)</b>
<b>Dark defects</b>	20	21	12	16	15
<b>Bright defects</b>	6	7	4	6	5
<b>Ratio</b>	3.33	3	3	2.66	3

Table 1.1: Table of defect counts

70 dark defects and 24 bright defects using the **Template Matching** method. In contrast, using the **FFT** method, we obtain the same number of bright defects but slightly fewer dark defects ( $\sim 65$ ). This discrepancy can be explained by the FFT method's tendency to count defect clusters as single defects.

#### 1.4.1 Defect Density Calculation

Now that we have established that the **bright defects** are caused by **Zr vacancies**, we can refine our **density calculations**. The defect densities of the **dark and bright defects** are computed relative to the number of **Te** and **Zr** ions present in the scanned area.

For a  $45 \times 45 \text{ nm}^2$  topography, we divide each axis by the unit cell dimensions as follows:

For the *a*-axis:

$$\frac{45 \text{ nm}}{0.58 \text{ nm}} \sim 77 \text{ atoms}$$

Multiplying by 2 to account for both Te(1) and Te(2) ions:

$$77 \times 2 \sim 155 \text{ Te ions}$$

For the *b*-axis:

$$\frac{45 \text{ nm}}{0.39 \text{ nm}} = 115 \text{ atoms}$$

Total number of Te ions:

$$155 \times 115 \sim 17,900 \text{ Te ions}$$

Density of dark defects with respect to Te ions:

$$\frac{70 \text{ (dark defects)}}{17,900 \text{ (total Te ions)}} = 0.40\%$$

For Zr ions:

$$\frac{45 \text{ nm}}{0.58 \text{ nm}} \sim 77, \quad \frac{45 \text{ nm}}{0.39 \text{ nm}} \sim 115$$

Total number of Zr ions:

$$77 \times 115 \sim 8,900 \text{ Zr ions}$$

Density of bright defects with respect to Zr ions:

$$\frac{24 \text{ (bright defects)}}{8,900 \text{ (total Zr ions)}} \sim 0.27\%$$

To validate our calculations, we also used MATLAB to detect high-intensity peaks and compare results and count each atom within the images.

If we do not differentiate between defect types, the total defect count is 94. The total defect density with respect to Te ions is:

$$\frac{94}{17,900} = 0.53\%$$

This result is consistent with a previous **STM study** on undoped  $\text{ZrTe}_3$  [8]. However, we believe that our study considers different types of defects compared to the aforementioned work. In their study, they counted **point defects** and suggested that these defects correspond to **missing Te ions**.

In summary, we calculated the densities of the defects observed in our topographies of  $\text{Zr}_{0.95}\text{Hf}_{0.05}\text{Te}_3$ . The **dark defects**, which are essentially **Te surface vacancies**, have a density of **0.4%**, while the **bright defects**, which we identify as **Zr vacancies**, have a density of **0.27%**.

These values are clearly not consistent with the expected **5% Hf doping**, suggesting that the observed defect densities may originate from additional factors.

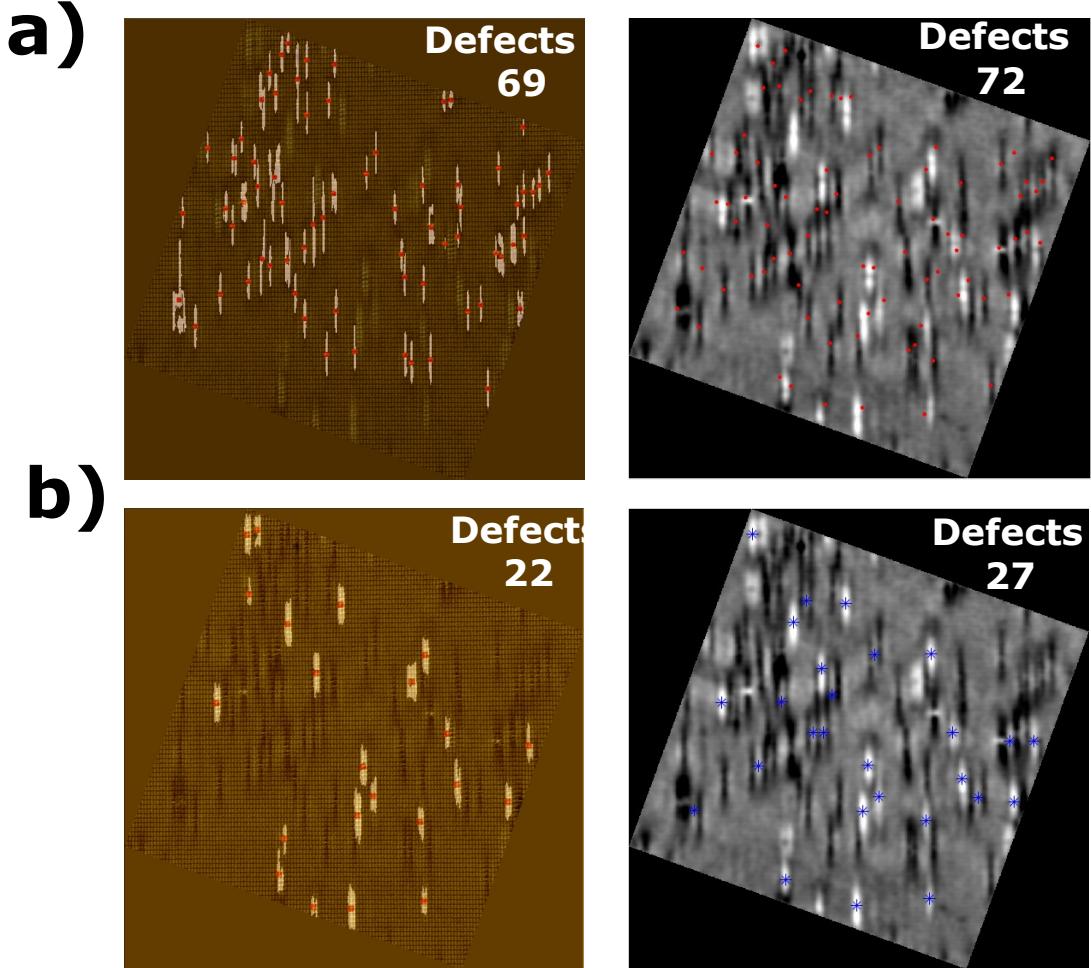


Figure 1.11: FFT vs. Template Matching method for defect counting. (a) Comparison for dark defect counting. (b) Comparison for bright defect counting.

#### 1.4.2 Defect Formation Energy in $\text{ZrTe}_3$ ( $7 \times 3$ Supercell)

In our topographies, we observe **dark defects** more frequently than **bright defects**, indicating that **Te vacancies** are more prominent than **Zr vacancies**. To relate this observation to the formation energy of each defect, we conducted **DFT calculations** on a  $7 \times 3$  supercell for  **$\text{ZrTe}_3$** , determining the following defect formation energies:

- Energy required to create a **Te surface vacancy**: 29 Ry (394 eV).
- Energy required to create a **Zr vacancy**: 100 Ry (1360 eV).

This indicates that a Zr vacancy requires  $\sim 3.4$  times more energy to form than a Te surface vacancy. Notably, the formation energy for a Te surface vacancy and a

subsurface vacancy is approximately the same.

### Defect Formation Energy in Hf-Substituted ZrTe<sub>3</sub> (7×3 Supercell)

For Hf-substituted ZrTe<sub>3</sub>, the defect formation energies are slightly higher:

- Energy required to create a **Te surface vacancy**: 55 Ry (748 eV).
- Energy required to create a **Zr vacancy**: 125 Ry (1700 eV).

The energy ratio for Zr defects vs. Te defects is approximately  $\sim 2.27$  in this case.

### Comparison to Experimental Defect Densities

Overall, the **defect formation energy** for both the pristine supercell and the **Hf-substituted supercell** indicates that the energy required to create a **Te vacancy** is significantly lower than that required to form a **Zr vacancy**.

Furthermore, our topographic analysis shows that **Te vacancies** are more prevalent than **Zr vacancies**, with respective densities of **0.4%** and **0.27%**, which is consistent with these calculations.

## 1.5 Cross Correlation Between Defects and CDW State in ZrTe<sub>3</sub>

One way to quantitatively examine the interplay between defects and the **CDW state** is by computing the **cross-correlation** between the defect signal and the CDW signal.

The **cross-correlation** is a mathematical method used to evaluate the degree of similarity between two signals based on their spatial or temporal displacement. In this study, it allows us to quantify the relationship between **defect locations** and the **CDW** maxima both below and above  $T_{\text{CDW}}$ .

To perform this analysis, we represent both the defect signal and the CDW maxima signal as binary masks and compute their **cross-correlation** using two approaches: the **Template Matching** method and the **FFT** method. We implemented these calculations using a MATLAB script, which computes the cross-correlation between two text files. The simplest application of this script involves reading a defects text file and a CDW maxima text file, then performing the cross-correlation using the `normxcorr2` function.

Additionally, instead of limiting the analysis to the CDW maxima, we extend our approach to compute the cross-correlation between the **center of the defects** and

the entire CDW signal. Overall we calculated the cross-correlation calculations across different temperatures using three distinct approaches:

- Cross-correlation between defects and CDW maxima.
- Cross-correlation between defects and the entire CDW signal.
- Cross-correlation between defect signals extracted from FFT and CDW maxima.

It is important to note that the CDW maxima are defined by masking 60% of the intensity in all images.

## Function `normxcorr2` Description

The function `normxcorr2` [9] follows the general procedure:

1. Calculate cross-correlation in the spatial or frequency domain, depending on the size of the images.
2. Calculate local sums by precomputing running sums.
3. Use local sums to normalize the cross-correlation to obtain correlation coefficients.

The implementation closely follows the formula:

$$r(u, v) = \frac{\frac{1}{n} \sum_{x,y} [f(x, y) - \bar{f}_{u,v}] [t(x - u, y - v) - \bar{t}]}{\sqrt{\sum_{x,y} [f(x, y) - \bar{f}_{u,v}]^2 \sum_{x,y} [t(x - u, y - v) - \bar{t}]^2}} \quad (1.1)$$

where:

- $n$  is the number of pixels.
- $f$  is the image.
- $t$  is the template.
- $\bar{f}_{u,v}$  is the mean of  $f(x, y)$  in the region under the template.
- $\sum_{x,y} [f(x, y) - \bar{f}_{u,v}]^2$  is the standard deviation of the image.
- $\sum_{x,y} [t(x - u, y - v) - \bar{t}]^2$  is the standard deviation of the template.

Below, Figure ?? presents the cross-correlation calculation for each image, where we illustrate the pinpointing of dark defects and their overlay on Gaussian-treated images to highlight the defect locations. On the right, we overlay the defect locations as red points over the CDW signal.

Notably, above the transition temperature, the red dots predominantly appear on the CDW maxima. Additionally, we display the number of defects detected using the template matching method.

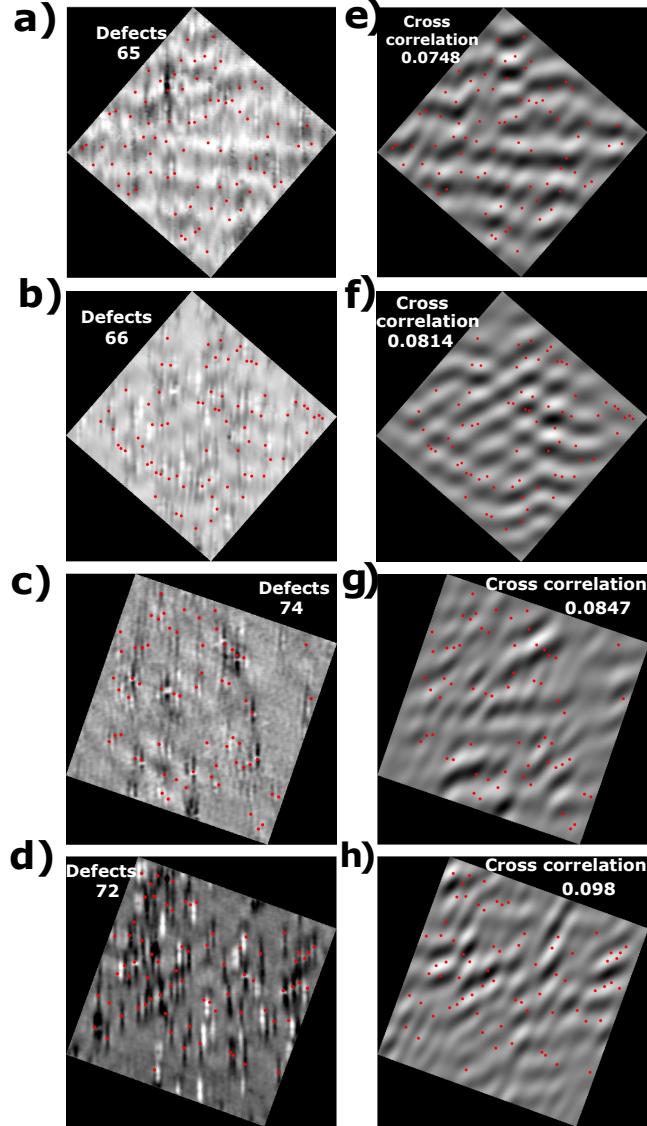


Figure 1.12: Dark defects overlay on CDW. (a-d)  $45 \times 45 \text{ nm}^2$  images taken at different temperatures: 90K, 80K, 60K, and 13K, respectively, under the same scanning conditions ( $V = -50 \text{ mV}$ ,  $I = 50 \text{ pA}$ ), with overlaid red dots marking dark defect locations. (e-h) CDW filtered signal overlaid with defect centers marked by red dots.

Here, we calculate the cross-correlation between the bright defects and the CDW

signal (Figure 1.13). Notably, the number of bright defects is considerably lower than that of dark defects, with an average ratio of approximately 3:1. This ratio is consistent with the formation energy discussed previously, where the energy required to create a Zr vacancy is higher than that required to create a Te vacancy.

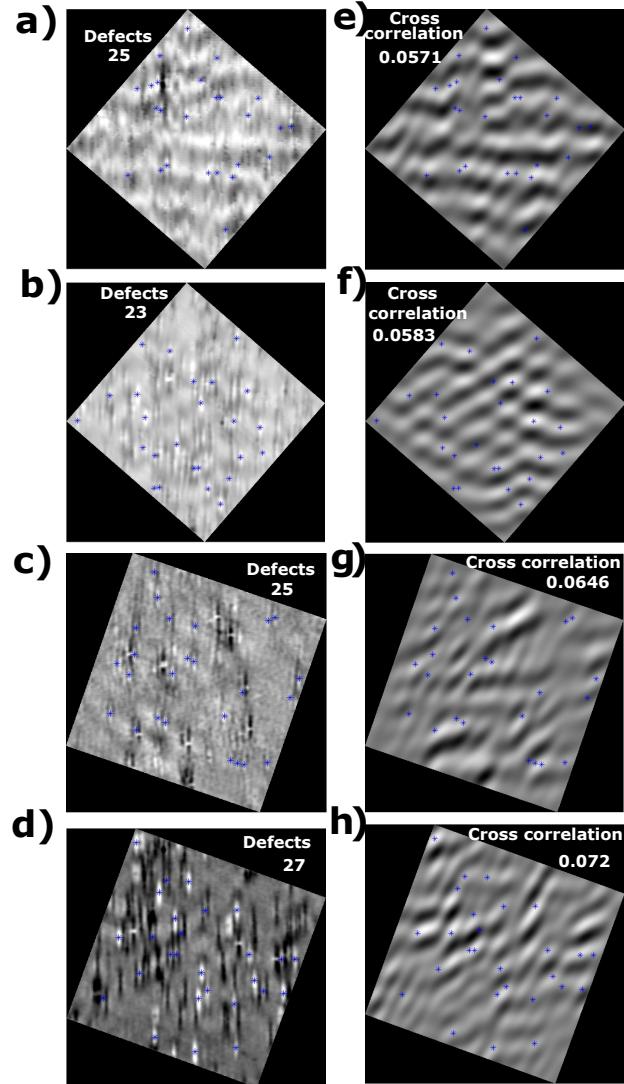


Figure 1.13: Bright defects overlay on CDW. (a-d)  $45 \times 45 \text{ nm}^2$  images taken at different temperatures: 90K, 80K, 60K, and 13K, respectively, under the same scanning conditions ( $V = -50 \text{ mV}$ ,  $I = 50 \text{ pA}$ ), with overlaid blue stars marking bright defect locations. (e-h) CDW filtered signal overlaid with defect centers marked by blue stars.

Finally, we compared the cross-correlation across different temperatures using three different approaches:

- Cross-correlation between defects detected via template matching and CDW maxima.
- Cross-correlation between defects detected via FFT and CDW maxima.
- Cross-correlation between defects detected via template matching and the entire filtered CDW signal.

All of the plots exhibit a similar trend, where the **cross-correlation** increases with temperature as the system transitions from below to above the **CDW transition temperature**. This suggests that the **CDW** becomes pinned to defect locations above the transition temperature, and this pinning effect increases with temperature for both **bright defects** (Zr vacancies) and **dark defects** (surface Te vacancies).

This result is consistent with previous **STM study[8]**, where the authors investigated **point-like defects**, believed to be Te vacancies, in relation to the **CDW maxima**. Their findings proposed a weak-to-strong pinning effect as the system moves from below to above  $T_{CDW}$ .

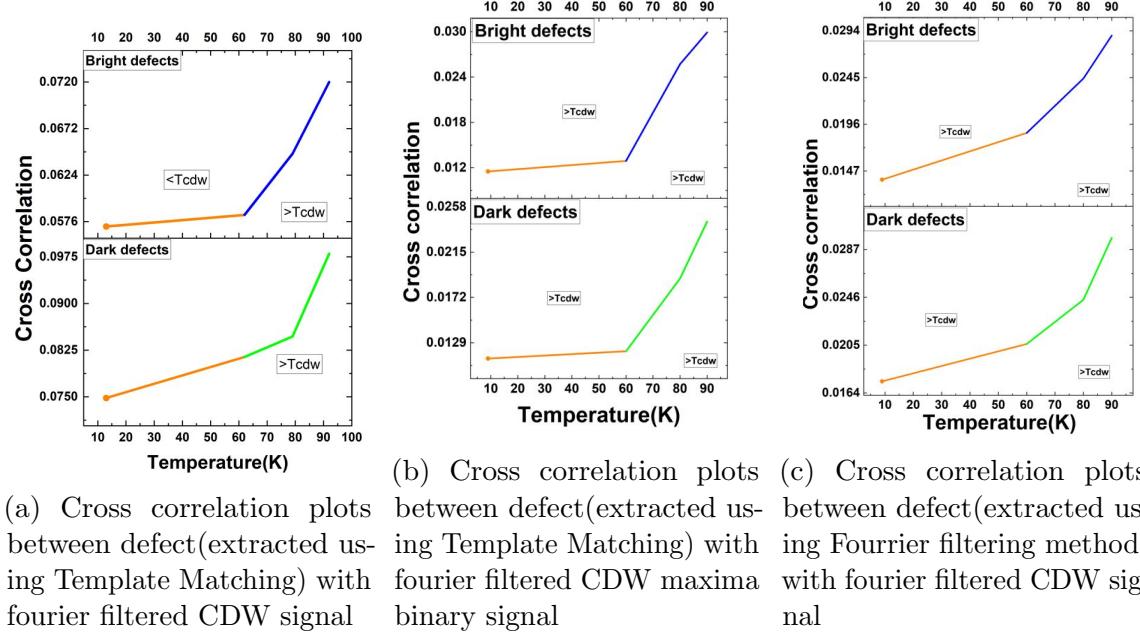


Figure 1.14: Temperature evolution of CDW using different correlation methods

## 1.6 Conclusion

We conducted temperature-dependent measurements, revealing that below the transition temperature, the CDW is observed uniformly across the topographies. Near the CDW transition, at 62 K, we noticed a pronounced waviness of the CDW. Furthermore, above the transition temperature, the **CDW** becomes even more wavy and appears localized around defect sites. These observations provide evidence of CDW pinning to defects and highlight the need to investigate the interplay between defects and the **CDW state** as the temperature increases across the **CDW transition**.

To achieve this, we computed the **cross-correlation** between defect locations and the **CDW signal**. The CDW signal was extracted from the **Fast Fourier Transform (FFT)**. However, determining defect locations required additional steps.

We employed three methods to extract the defect signal:

1. Fast Fourier Transform (FFT) Method: This method filters a rectangular area in the low-wave-vector region to isolate defect-related features.
2. Template Matching Method: This technique computes the cross-correlation between a defect template and the entire topography to pinpoint similar defect features.
3. Machine Learning Approach: We utilized **YOLOv8**, a deep learning tool, to locate defects within the topography.

We applied these methods to calculate defect densities and found that the **dark defects** (Te surface vacancies) have a density of **0.4%** with respect to **surface Te ions**, while the **bright defects** (Zr vacancies) have a density of **0.27%** with respect to **Zr ions**. These results align with our **DFT calculations**, which show that the energy required to create a **Zr vacancy** is significantly higher than that required to create a **Te surface vacancy**.

Finally, the computed **cross-correlation** between each defect and the **CDW signal**, , **increases with temperature**, indicating a potential **pinning mechanism** between **bright and dark defects** and the **CDW state**. This pinning effect strengthens as the system transitions from below to above the **CDW transition temperature**.

## References

- [1] Mahmoud Jameel Atta Daasan and Mohamad Hafis Izran Bin Ishak. Enhancing face recognition accuracy through integration of yolo v8 and deep learning: A custom recognition model approach. In Fazilah Hassan, Noorhazirah Sunar, Mohd Ariffanan Mohd Basri, Mohd Saiful Azimi Mahmud, Mohamad Hafis Izran Ishak, and Mohamed Sultan Mohamed Ali, editors, *Methods and Applications for Modeling and Simulation of Complex Systems*, pages 242–253, Singapore, 2024. Springer Nature Singapore.
- [2] Juan Terven, Diana-Margarita Córdova-Esparza, and Julio-Alejandro Romero-González. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction*, 5(4):1680–1716, 2023.
- [3] Tianyong Wu and Youkou Dong. Yolo-se: Improved yolov8 for remote sensing object detection and recognition. *Applied Sciences*, 13(24), 2023.
- [4] Christine Dewi, Danny Manongga, Hendry, Evangs Mailoa, and Kristoko Dwi Hartomo. Deep learning and yolov8 utilized in an accurate face mask detection system. *Big Data and Cognitive Computing*, 8(1), 2024.
- [5] Yinguang Zhang, Jianhuan Su, Teng Wang, Chuan Xu, and Ao Yu. Intelligent detection method of microparticle virus in silkworm based on yolov8 improved algorithm. *The Journal of Supercomputing*, 80(12):18118–18141, August 2024.
- [6] Ultralytics. Ultralytics official website, 2024.
- [7] Roboflow. Roboflow: Give your software the eyes of a developer, 2025. Accessed: 2025-04-12.
- [8] Limin Liu, Changjiang Zhu, Z. Y. Liu, Hanbin Deng, X. B. Zhou, Yuan Li, Yingkai Sun, Xiong Huang, Shuaishuai Li, Xin Du, Zheng Wang, Tong Guan, Hanqing Mao, Y. Sui, Rui Wu, Jia-Xin Yin, J.-G. Cheng, and Shuheng H. Pan. Thermal dynamics of charge density wave pinning in zrte<sub>3</sub>. *Phys. Rev. Lett.*, 126:256401, Jun 2021.
- [9] J. P. Lewis. Fast normalized cross-correlation. Online, 1995. Accessed: [Insert Date].