# Employing CycleGANs to Generate Realistic STEM Images for Machine Learning


**Abid Khan**
Department of Physics
University of Illinois at Urbana-Champaign
Urbana, IL 61801
aakhan3@illinois.edu

**Chia-Hao Lee**
Department of Materials Science and Engineering
University of Illinois Urbana-Champaign
Urbana, IL 61801
chiahao3@illinois.edu

**Pinshane Y. Huang**
Department of Materials Science and Engineering
University of Illinois Urbana-Champaign
Urbana, IL 61801
pyhuang@illinois.edu

**Bryan K. Clark**
Department Physics
University of Illinois Urbana-Champaign
Urbana, IL 61801
bkclark@illinois.edu


## Abstract


Identifying atomic features in aberration-corrected scanning transmission electron microscopy (STEM) data is critical to understanding structures and properties of materials. Machine learning (ML) models have been applied to accelerate these tasks. The training sets for these ML models are typically constructed with codes that provide simulations of STEM images alongside desired labels. However, these simulated images are often limited by the oversimplified model and deviate from the experimental images, limiting the accuracy and precision of ML training. We present an approach to generating realistic STEM images by employing a cycleGAN to automatically add realistic microscopy features and noise profiles to simulated data. We also train a defect-identification neural network using these generated images and evaluate the model on real STEM images to locate atomic defects within them. The application of CycleGAN provides other machine learning models with more realistic training data for any type of supervised learning.


## 1 Introduction

Historically, experimental data is acquired and then analyzed in a way that often requires significant human input. In the context of scanning transmission electron microscopy (STEM) data, this typically involves identifying aspects of the material such as atom locations, atom types, and atomic defects by hand. Recently, Lee et al. [2020] automated the process of identifying atomic defects on massive datasets using machine learning techniques, where a very large number of single-atomic defects were collected to measure their induced strains at a sub-picometer precision. These measurements prove to be vital in understanding the structure and properties of 2D materials, and machine learning is critical to accomplish this at the scale necessary to collect the requisite statistics.


36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Training a neural network to automatically label image properties such as defect locations requires a large training set of data. While high accuracy training data can be generated directly from experimental data, this partially defeats the purpose of using ML in the first place as a domain expert is needed for this labelling. To attempt to get around this, we generate artificial data as a training set from an experimental simulator. Unfortunately, the output of such simulators differs from experiment, typically requiring simulated data to be digitally altered, often through manual user input, to replicate experimental data. It is this problem we will address in the context of STEM in this paper. In particular, we develop a CycleGAN which takes simulated data and modifies it so as to be largely indistinguishable from actual experimental images. We proceed to demonstrate the efficacy of our cycleGAN by using it to generate training data for a fully convolutional neural network (FCN) to find single-atomic defects in experimental STEM images.

## 2 The Data: Experimental STEM Images vs Simulated Images

For a cycleGAN, we require two sets of images to style-transfer between. The first set of images is experimentally obtained STEM images. The second set is simulated STEM images. For the experimental images, we acquire aberration-corrected annular dark-field (ADF) STEM images of three materials: graphene, $WSe_2$ , and $SrTiO_3$ (STO). We particularly focus on the acquisition of $WSe_2$ , where capturing images occurred on two days, acquisition day A and day B forming two datasets; we include a third dataset AB which includes both days of STEM images.

We used Incostem, an incoherent STEM image simulation software bundled in the CompuTEM package to obtain the simulated STEM image which corresponds to a ‘pristine‘ image of the experimental material (see Kirkland [2022]). These simulation images come with labels (such as defect locations) that will not be seen by the cycleGAN but are critical for the training of other machine learning models and therefore must be preserved by the cycleGAN. Incostem can produce arbitrarily large training datasets.

The simulation images deviate non-trivially from the experimental images in that the simulations do not fully capture the microscopy conditions or the background contamination in the material. Such deviations include detector noise, electrical noise, scan distortion, sample drift, surface contamination, damage, and aberration (see Madsen et al. [2018]). A cycleGAN transforms simulation images to automatically incorporate these realistic microscopy and material conditions. As explained by Zhu et al. [2017], cycleGANs are often used in image processing and have been successful in transferring the style of one image set to another such as converting a photographed image into a "Monet"-like painting. The goal of our cycleGAN, then, is to perform a "style transfer" from a dataset of experimental STEM images to a dataset of simulated images.

## 3 The cycleGAN

It is important that our images not only look experimentally realistic but also that the labels corresponding to feature locations stay stable - i.e. it is a failure if the generator scrambles the location of all of the features from the simulated image. CycleGANs add significant machinery on top of standard GANs to resolve this problem by ensuring that the local features, and hence labels are preserved. The cycle-consistency loss and the identity loss serve as necessary regularization terms to enforce the preservation of these local features.

In most common datasets of images, almost all of the information is captured in its real-space image. Atomic scale images of crystals, however, exhibit significant periodicity, and so momentum space information is highly relevant. The Fourier space of these images must also look realistic then, so we add two discriminators in addition to the conventional cycleGAN described by Zhu et al. [2017] that distinguish the amplitude of Fourier transforms of both simulated images and realistic experimental images. Figure 1 illustrates the full cycleGAN model that is implemented. Each generator is then trained to not only fool a real-space image discriminator, but also a momentum-space discriminator.

Khan et al. [2022] provides the full code for the cycleGAN along with an FCN model to detect single-atomic defects. The generators follow a U-net architecture, while the discriminators are patch-level as described by Zhu et al. [2017]. To prepare the datasets for training, for each dataset we normalize all the 1024x1024 images and cut them into 256x256 pieces. The batch-size is 32, however when adding each piece into the batch, they are randomly rotated by a multiple of 90 degrees, and they

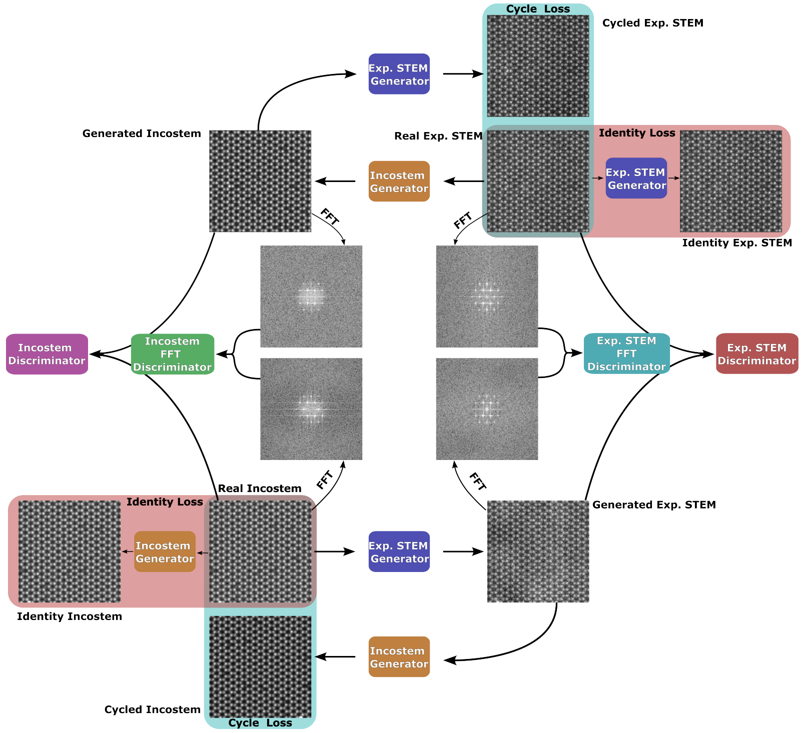


Figure 1: Schematic of our cycleGAN. Along with the conventional two generators and two discriminators, we have two additional discriminators for the FFTs of the experimental STEM images. We show the cycle-consistency (identity) loss as minimizing the difference between the vertical (horizontal) images grouped in the cyan (magenta) box.

are also sometimes inverted. The simulated images initially contain no noise, which is detrimental to a cycleGAN because a sufficient amount of variability is required to generate unique images. We add to the "pristine" simulated images, Gaussian noise to ensure enough variability across the simulation dataset. Each pixel in the image is adjusted by a Gaussian distribution of $\mathcal{N}(0, 0.1)$. All networks were trained for 200 epochs, where in the first 100 epochs, the learning rate was 0.002, and from epochs, 100-200, the learning rate linearly decayed to zero. For the cycle-consistency loss, $\lambda$ was 10, and for the distortion loss, $\lambda$ was 5. These hyperparameters were determined after substantial tuning and testing, where we trained an FCN to find defects in $WSe_2$ , and chose the cycleGAN model that provided the best results.

We use Google's Tensorflow backend with the National Center for Supercomputing Application's (NCSA) Delta cluster. We train a cycleGAN for 9 datasets, and use 1 NVIDIA A40 GPU per training. Training a single cycleGAN for 200 epochs takes about 6 hours of compute time.

The main objective of our cycleGAN is to develop a generator which turns simulated images into experimental-like data; given such a device, we can then use it as a source to generate an arbitrarily large amount of training data by generating simulated data which is then processed by the cycleGAN to make it appear experimental. We apply our cycleGAN in this way to detect single-atom defects in experimentally obtained STEM images of an alloyed 2D transition metal dichalcogenide (TMDC) monolayer $WSe_2$ . First we acquire experimental and simulated STEM images of the material that contains defects; importantly the simulated STEM images also contain labelled defect locations. Given these two datasets, we train a cycleGAN to perform "style-transferring" between them providing us with a generator that turns simulated STEM images into realistic-looking experimental STEM images. Using this generator, we construct a dataset of generated experimental STEM images. We take this dataset alongside their corresponding atomic defect locations, and train a fully convolutional neural network (FCN) to locate defects in the generated experimental STEM images. With this FCN trained, we can feed it real experimental STEM images to locate where their defects lie. Note that the efficacy of this process validates that the defect labelling is largely preserved.

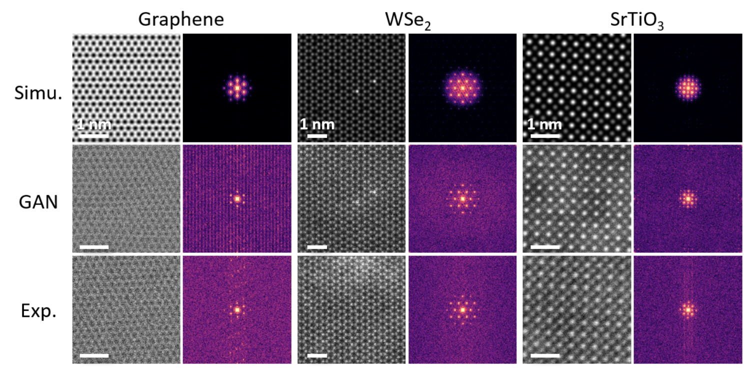


Figure 2: Prototypical images of graphene (left two columns), $WSe_2$ (middle two columns), and $SrTiO_3$ (right two columns). Each column displays the real space image along with its Fourier transform next to it. For each material, we show an image from Incostem (top row), its generated STEM image after passing through a cycleGAN (middle row), and a prototypical experimental STEM image (bottom row) for comparison.

## 4 Results

Figure 2 show cycleGAN results of Graphene, $WSe_2$ , STO. Qualitatively, we see that both the real-space and momentum-space generated images are nearly indistinguishable to the experimental STEM images even to domain experts and both are relatively distant from the original simulated images.

To get a more quantitative evaluation of the images generated from cycleGANs, we look at how well they work in defect identification training. Using the generated STEM images of $WSe_2$ along with the given locations of single chalcogen-site vacancies, we train a res-unet FCN to locate Se vacancies in experimental STEM images of $WSe_2$ . The experimental dataset of $WSe_2$ is divided into two datasets: day A and day B. In each of these days, we manually labeled single-vacancy defects in 3 1024x1024 images so as to validate our results. We then evaluate our FCN on these images, and compare the predicted single-vacancy locations with the manually labeled single-vacancy locations.

We train a set of FCNs using different initial training data. The first training dataset we use is simulation data without any noise - i.e. Incostem data without any processing. The next few training sets are generated via cycleGAN on datasets A, B, and AB, respectively. We use the same set of experimental images for both training the cycleGAN as well as for labelling defects; this ensures the cycleGAN correctly identifies the microscopy conditions that are present (which drift over time) in that particular set of images that we are labeling. Table 1 shows the precisions, recalls, and F1 scores of the FCN given the training dataset that was used.

There are two key take-aways of these results. First, we find adding noise into the simulated dataset is necessary for the FCN to locate defects in experimentally obtained STEM images. The F1 scores for the noiseless datasets perform relatively poorly. When adding noise from a cycleGAN, the FCN performs significantly better. Secondly, we ask the question, how much data does one want to use to train a GAN. There is an inherent tension here. The more data, the more robust the GAN will be. On the other hand, over time microscopy and material conditions change and so training a GAN on a set of images taken over a longer stretch of time will result in images that are not as perfectly matched with the given image you are trying to train on. We find, in our analysis, that the sweet spot appears to be to train over single days of data (100-200 1K-images).

## 5 Conclusion

We successfully construct a cycleGAN model to generate realistic experimental STEM images from simulation. This was done by ensuring that the generators not only make the real-space images look realistic, but also its momentum space. We demonstrate the ability for this generator to construct training data by training an FCN to find single-atom defects trained with data generated from a

Table 1: Precisions (P), Recalls (R), and F1 Scores (F1) of FCN defect-identification models.

| | Day A | | | Day B | | | Day AB | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | **F1** | P | R | **F1** | P | R | **F1** |
| no noise | 15 | 83 | **25** | 66 | 84 | **74** | 22 | 83 | **35** |
| cycleGAN | 85 | 79 | **82** | 87 | 91 | **89** | 89 | 63 | **74** |

cycleGAN. While this technique was meant for generating training data for STEM ML models initially, we can extend this idea to any supervised learning problem where real data lacks labels, but there exists an efficient simulation that construct input data with labels but not necessarily with realistic experimental conditions.

## 6 Broader Impacts

Employing machine learning has been shown to be highly instrumental in understanding the physics of materials. While the ability to construct realistic-looking experimental STEM images is critical in training ML architectures to analyze scientific data at scale, there are concerns, most clearly being generating fake STEM images and having them be passed off as real scientific data. These cycleGANs that have been constructed have the ability to generate STEM images that may seem indistinguishable from real STEM images. One could then construct simulation images that are tailored to an authors' theory, pass it through a trained cycleGAN, producing "experimentally-obtained" STEM images, and show that experiment confirms the theory. While this may be concerning, it also drives us to study more quantitatively the differences between real and generated STEM images and is a part of a broader societal challenge of distinguishing real from computer-generated images.

## References


Abid Khan, Di Luo, and Chia-Hao Lee. Stem-learning, 2022. URL `https://github.com/ClarkResearchGroup/stem-learning`.

Earl J. Kirkland. computem, 2022. URL `https://sourceforge.net/projects/computem/`.

Chia-Hao Lee, Abid Khan, Di Luo, Tatiane P. Santos, Chuqiao Shi, Blanka E. Janicek, Sangmin Kang, Wenjuan Zhu, Nahil A. Sobh, André Schleife, Bryan K. Clark, and Pinshane Y. Huang. Deep learning enabled strain mapping of single-atom defects in two-dimensional transition metal dichalcogenides with sub-picometer precision. Nano Letters, 20(5):3369–3377, 2020. doi: 10.1021/acs.nanolett.0c00269. URL `https://doi.org/10.1021/acs.nanolett.0c00269`. PMID: 32243178.

Jacob Madsen, Pei Liu, Jens Kling, Jakob Birkedal Wagner, Thomas Willum Hansen, Ole Winther, and Jakob Schiøtz. A deep learning approach to identify local structures in atomic-resolution transmission electron microscopy images. Advanced Theory and Simulations, 1(8):1800037, jul 2018. doi: 10.1002/adts.201800037. URL `https://doi.org/10.1002%2Fadts.201800037`.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. CoRR, abs/1703.10593, 2017. URL `http://arxiv.org/abs/1703.10593`.


## Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] We discuss issues of dataset sizes in section 4.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 6

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

   (a) Did you state the full set of assumptions of all theoretical results? [N/A] No theoretical claims are made.

   (b) Did you include complete proofs of all theoretical results? [N/A] No theoretical proofs are made.

3. If you ran experiments...

   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See section 3

   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See section 3

   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] For all experiments, only one seed was run

   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See section 3

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

   (a) If your work uses existing assets, did you cite the creators? [Yes] For the FCN model, we've cited Lee et al. [2020]

   (b) Did you mention the license of the assets? [N/A] No licenses assets were used.

   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See section 3

   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] All data was created in-house.

   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] The data used here is clearly inoffensive.

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] No crowd sourcing or human subjects were employed.

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]