Towards Differentiable Electron Microscopy Simulation: Methods and Applications

Ngan Nguyen*
King Abdullah University of Science and Technology
Feng Liang†
King Abdullah University of Science and Technology
Dominik Engel‡
Ulm University
Ciril Bohak§
King Abdullah University of Science and Technology
Ondřej Strnad¶
King Abdullah University of Science and Technology
Timo Ropinski‖
Ulm University
Ivan Viola**
King Abdullah University of Science and Technology

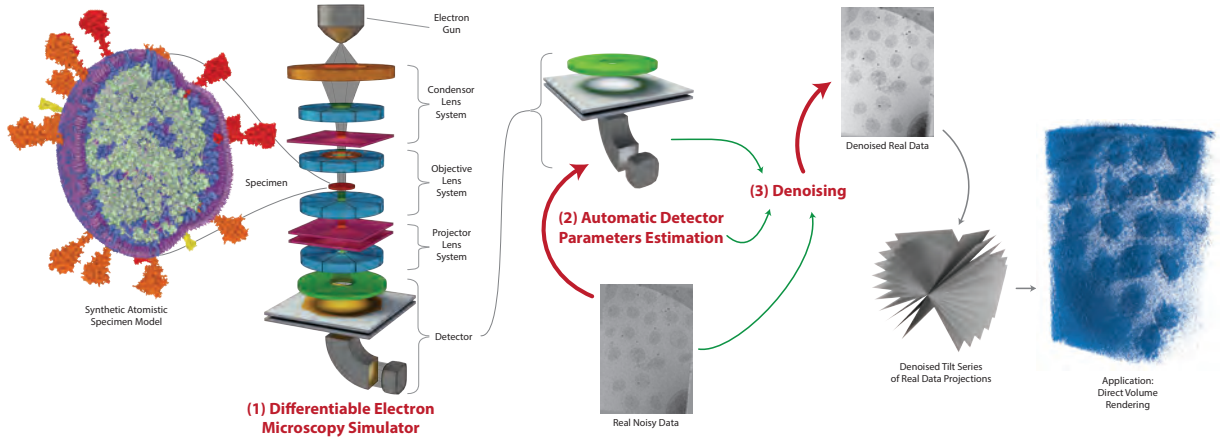

Figure 1: The three main contributions of this work are: (1) DiffTEM: a differentiable electron microscopy simulator, that (2) facilitates automatic detector parameter estimation, and (3) can be used for denoising the data from real electron microscopes. The gray arrows point toward output. The green arrows show the flow of optimization input. The red arrows show the flow of optimization and back propagation.

Abstract

We propose a new microscopy simulation system that can depict atomistic models in a micrograph visual style, similar to physical electron microscopy imaging results. This system is scalable, able to represent the simulation of electron microscopy of twenty complex viral particles, and synthesizes the image faster than previous approaches. Additionally, the simulator is differentiable in the deterministic and stochastic stages that form signal and noise representations in the micrograph. This allows for solving inverse problems by means of optimization and thus allows for the generation of microscopy simulations using parameter settings estimated from real data. We demonstrate this capability through two applications: (1) estimating the parameters of the modulation transfer function defining the detector properties of the simulated and real micrographs and (2) denoising the real data based on parameters optimized from the simulated examples using gradient descent. While current simulators do not support any parameter estimation due to their forward-only design, we show that the results obtained using estimated parameters are very similar to real micrographs. Additionally, we evaluate the denoising capabilities of our approach and show that the results are competitive with state-of-the-art methods. Denoised micrographs exhibit less noise in the tomographic reconstruction of tilt-series, facilitating visualization of microscopy tomography using direct volume rendering by reducing the visual dominance of noise.

Index Terms: Computing methodologies—Modeling and simulation—Simulation types and techniques—Scientific visualization; Computing methodologies—Modeling and simulation—Simulation support systems—Simulation tools;

1 Introduction

With the revolution in resolution [23] in 2014, cryogenic electron microscopy (cryo-EM) became a unique tool to study and visualize the cell and its inner structures at the molecular level [32]. Still, extracting valuable information from the data obtained using cryo-EM and developing the computational methods for processing, analyzing, and/or visualizing such data is non-trivial. The main reason is that the data acquisition process is time-consuming, costly, and still mainly performed by bio-experts. This situation results in a relatively small amount of data available for developing learning-based computational methods. Additionally, due to the limitation to only use low electron beam doses to protect the specimen, the data always suffer from a low signal-to-noise ratio (SNR). Finally, because of the limited specimen tilting during the acquisition pro-

*e-mail: ngan.nguyen@kaust.edu.sa, shared first authorship
†e-mail: feng.liang@kaust.edu.sa, shared first authorship
‡e-mail: dominik.engel@uni-ulm.de
§e-mail: ciril.bohak@kaust.edu.sa
¶e-mail: ondrej.strnad@kaust.edu.sa
‖e-mail: timo.ropinski@uni-ulm.de
**e-mail: ivan.viola@kaust.edu.sa

2024 IEEE 17th Pacific Visualization Conference (PacificVis) | 979-8-3503-9380-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/PacificVis60374.2024.00016

cess, high-tilt projections are missing, which results in the missing wedge problem [39]. All these shortcomings make the usage and visualization of such data difficult. To overcome these challenges, it is possible to use highly-accurate simulated data, which are close to indistinguishable from actual data in ground truth evaluations since the input structural model is known before the microscopy simulation process. It can also be used to train structural biologists to interpret microscopy projections, i.e., micrographs. Additionally, such simulation can also be regarded as a distinct *rendering style*, i.e.physically-based rendering of the electron-microscopy data, a wave-based rendering approach based on Fourier Optics theory, and as such, it can be useful in generating animation sequences where a biological structure is transformed into a micrograph and vice versa. Last but not least, such data can be used for developing new Deep Learning (DL) methods for interpreting real micrographs, their testing, evaluation, validation, and visualization.

In recent years, researchers have developed several Transmission Electron Microscopy (TEM) simulators to support studies in biological specimen research, some of which are based on physical principles [13,36,41] or parallel multi-slice simulation [28]. The simulator presented by Rullgård *et al.* [36] uses the multi-slice method [5] for modeling the interaction of electron and biological specimen, which is well suited for a computer simulation [20]. Additionally, the authors have developed a phantom generator for generating a specimen model and calculating its scattering potential based on a physical model. The above-mentioned simulator properties make it a good choice for simulating an acquisition process of biological specimens using cryo-EM. However, the simulator has two major limitations. First, the simulated microscope parameters must be calibrated manually, which is a tedious and time-consuming process where even the initial values are unavailable. Second, the simulator is very slow when simulating structures larger than a few proteins. Some enveloped viral particles are huge macromolecular models (e.g., HIV or SARS-CoV-2). In the experiments, the specimen often contains a high number of instances of such particles.

In the field of differentiable rendering, visualization problems are addressed using differentiable pipelines. For example, DiffDVR [43] uses a differentiable direct volume rendering (DVR) pipeline to optimize viewpoints, transfer functions, and voxel properties. Therefore, taking inspiration from differentiable rendering, we present a fast differentiable simulator, which enables us to optimize for all continuous detector parameters of a electron microscope. We demonstrate the capabilities of our differentiable simulator in two use cases, which are (1) detector parameter estimation from real data and (2) denoising. We show that using our differentiable simulator's detector parameters estimated from real data yields accurate results. Additionally, denoising real micrographs using our pipeline is effective and competitive with state-of-the-art DL-based approaches. Our implementation and optimization scripts are publicly available[1].

This paper presents several technical contributions:

- The overall architecture of the first differentiable microscopy simulator that is scalable to render realistic microscopy simulations within seconds or minutes.
- Present a wave-based rendering approach for electron-microscopy data, based on Fourier Optics theory.
- Estimating detector parameters from real projections to generate synthetic micrographs with the same detector characteristics that lead to a similar visual appearance of the synthetic and the real projection.
- Utilization of deterministic and stochastic differentiation for various stages of the microscopy simulation pipeline.
- A new denoising method that optimizes for a noise-free version of real-world noisy projections yielding noise-reduced tomographic reconstruction volumes suitable for 3D visualization.

[1] https://github.com/nanovis/diff-tem-release

2 Related work

Several Electron Microscopy (EM) simulators were developed for simulating the specimen acquisition process from different domains as presented in a book by Kirkland [20]. One of the most commonly used simulators for non-crystalline biological specimens is TEM simulator presented by Rullgård *et al.* [36], which is also the basis for our differentiable version of the TEM simulator. The TEM simulator aims to provide an accurate simulation based on a well-defined physical model. Still, the implementation of the simulator introduces a set of user-adjustable parameters (e.g., detector parameters), which require a time-consuming manual calibration process. The researchers have later presented TEM simulators that aim to improve the above one. While the original TEM simulator calculates the scattering potential of the specimen by estimating the potential of the atoms, InSilico TEM [41] calculates it more precisely by also including the scattering potential of molecular bonds. It also reduces the number of detector parameters and estimates them from experiments. In the recent work of Himes and Grigorieff [13], they considered the TEM simulation of amorphous samples sensitive to electron radiations. They also slice the simulation in the time domain, leading to the generation of movie frames of exposure. The latter approaches are more accurate than the original TEM simulator [36], trading in additional physical aspects of the simulator with much higher computational costs. In addition, when simulating large and intricate specimens using the multi-slice method, it is crucial to manage computational resources carefully. To address this challenge, Lobato and Van Dyck [28] proposed a parallel version of the multi-slice method using CUDA that allows performing extensive simulations that can include millions of atoms and require memory resources at a reasonable level. Additionally, some recent works and reviews [6, 19, 34] also proposed models and methods that can be used to solve inverse problems of cryo-EM, such as the reconstruction of an exit wave function. Kirkland and Meyer [19] reviewed the underlying theories of exit plane wave function reconstruction. They also illustrated how to use the theory in the indirect reconstruction of the exit wave function from the tilt series. In the work of Ophus and Ewalds [34], they used a simple analytic model of the iterative wave function to derive reconstruction guidelines. Donatelli and Spence [6] developed an iterated projection algorithm that reconstructs a scattering potential from N-beam multiple Bragg scattered intensities. These theories and methods have the potential for solving inverse problems, but all these works did not put their model in a complete simulator.

Nevertheless, all the presented simulators support only forward simulation. To the best of our knowledge, our proposed work is the first to offer a forward and backward approach to simulation through differentiability, which is crucial for the optimization of different tasks in the TEM domain, such as estimation of parameters for experiments, reconstruction of structures present in the specimen, etc.

Together with neural rendering, differentiable rendering has seen dramatic progress, mostly in the context of computer vision and photorealistic rendering. For example, in Li *et al.* [27], rendering parameters (e.g., scene geometries, camera intrinsics, and lighting conditions) can be estimated directly from images thanks to their differentiability. Hasselgren *et al.* [12] use a gradient estimator to optimize for ray tracing rendering parameters (i.e., number of samples per pixel). MVTN [11] takes advantage of a differentiable renderer to optimize for the performance of a downstream task (i.e., 3D geometry classification).

If we compare photorealistic rendering with TEM simulation in a general sense, they both share similar pipelines from geometries to rendered artifacts, given a rendering method. Similar to how differentiable rendering has opened a new view on solving well-known problems in a novel way with an explainable approach, can differentiable EM simulation provide new insight into solving

simulation-related problems with an optimization approach. CryoGAN [10] already shows promising results along these lines. The main difference is that it was developed for single-particle reconstruction and is thus lacking generalizability. In their recent work, Kniesel *et al.* [21] developed a differentiable Scanning Transmission Electron Microscopy (STEM) pipeline and showed promising results for implicit volume reconstruction. With our TEM simulator, we can estimate modulation transfer function (MTF) parameters of the simulator's detector part, which is in a way similar to estimating rendering parameters in a differentiable rendering method as was shown by Weiss and Westermann [43] for direct volume rendering. Moreover, our approach provides a novel approach for denoising by using images generated with a TEM simulator in a similar way as Noise2Noise [24] and Topaz-Denoise [1], but based on a simulated physical model [36].

Distribution learning has been a challenging problem in various fields of DL. There are many approaches to learning a random distribution. Still, they can be classified into two categories, learning the parameters of distributions (e.g., μ and σ of a normal distribution) and learning the manipulation of distributions, (e.g., $f_\theta(x)$ where θ parameterizes f, and x is sampled from a prescribed distribution). To address the problem of learning parameters of distributions, Schulman *et al.* [38] seek a general formulation of probabilistic models, which is stochastic compute graphs. Based on stochastic compute graphs, they devised a new way to transform deterministic loss functions into ones whose gradients are general gradient estimators of all parameters, including parameters of distributions. Such loss functions are called surrogate loss functions, and the compute graph, including surrogate loss functions, is called a surrogate loss function computation graph. In our work, to denoise micrographs, we define the surrogate loss function computation graph for gradient estimation based on the work of Schulman *et al.* [38].

When denoising cryo-EM micrographs, clean references are not available, so many approaches have been proposed to specifically denoise cryo-EM micrographs, addressing this problem.

Researchers have proposed various analytical filters for denoising, such as simple averaging, Gaussian filters, and many filters in the frequency space that enhance or dampen signal responses. Additionally, physics-based techniques were proposed, such as Nonlinear Anisotropic Diffusion (NAD) [8], which takes advantage of a diffusion model to denoise 3D volumes. For a comprehensive review on filters used in cryo-EM, we refer the reader to the survey by Huang *et al.* [16].

In the era of DL, many neural networks are proposed for denoising tasks. The first works in the denoising domain started by training neural networks to predict the clean image from a corrupted noisy image [29]. However, these works rely on the availability of clean ground truth to compare their predictions to form the loss, which is unavailable in the case of cryo-EM. Noise2Noise [24] is a general denoising approach that does not require clean data, enabling neural denoising in domains without access to ground truth. Instead of clean data, Noise2Noise is trained only on pairs of corrupted samples and learns the denoising task without the need for a likelihood model of the noise nor a density model for the clean data. Their approach is the first to realize that the distribution of clean images can be reconstructed from just noisy observations of that distribution. The resulting training framework utilizing only noisy images has subsequently been adapted in CryoCARE [4] and Topaz-Denoise [1]. In CryoCARE [4], a neural network is trained as a denoiser with multiple micrographs from multiple consecutive exposures or aligned micrographs in a tilt series or movie frames captured during a long exposure. In Topaz-Denoise [1], the researchers trained the model with a larger dataset containing thousands of micrographs acquired in different conditions, which enables better generalizability of their neural networks. Besides Noise2Noise-based approaches, Noise-Transfer2Clean (NT2C) [26] trains two neural denoisers and a Generative Adversarial Network (GAN)-based noise synthesizer to decouple noise from micrographs that are generated using a cryo-EM simulator based on real and synthetic data.

The implicit reconstruction approach by Kniesel *et al.* [21] also incorporates a normalizing flows neural network to model the noise distribution in the STEM data in a signal-dependent manner, allowing them to separate the noise from the clean reconstruction.

Our approach is similar to Noise2Noise [24] in the way that our model is required to output similar noisy images (i.e., noisy EM projections). However, our simulator does not involve neural networks but a physical model.

3 Differentiable simulator

Generally, to develop a TEM simulator, one critical step is determining the forward model that describes the image formation process. In brief, this forward model describes how the electron beam propagates through the specimen and forms an image at the image plane corresponding to the intensity recorded at a detector. We detail the physical foundations of this process in the Appendix (Appx.) A of the supplementary material.

Forward model In our work, the forward model is based on the one proposed by Rullgård *et al.* [36]. This model considers conventional bright-field TEM imaging of amorphous specimens, which is the prevalent imaging mode for cryo-EM tomography in the life sciences. Under restricted conditions, the bright-field TEM image can be modeled with a linear image model [20].

This model is designed similarly to the linear image model in light optics presented by Goodman [9]. This design has three main components: a source field (at the object plane), pupils, and a detector (at the image plane). The source field is a scattered wave field generated from a single incident electron traveling along the optical axis as it scatters against a specimen characterized by its scattering potential. The pupils are virtual apertures that indicate the *opening* available to collect scattering from the scattered wave field (entrance pupil) and the *opening* from which the collected scattering exits on its way to form an image (exit pupil).

In the next part, we present the differentiable parts we use to estimate detector parameters and denoise noisy micrographs. The detector records the image intensity or square magnitude of the image wave function $g(\boldsymbol{x})$, where $\boldsymbol{x} = (x, y)$ is a two-dimensional position vector in the image plane.

Noise Modeling In an ideal detection system, the final image is $g(\boldsymbol{x})$, but all detection systems introduce noise and distortions, so we need to model these components. The detector is defined as a rectangular area of square pixels and is characterized by parameters such as overall gain, blurring, and noise sources.

The primary type of noise is shot noise [37] due to fluctuations in the number of electrons that are emitted within a given time from the electron source and detected at each pixel. The electron detection in one pixel is independent of other pixels. Therefore, it is reasonably modeled with the Poisson distribution, taking the pixel intensity as the expected value. Let $C(i, j)$ denote the electron count at the (i, j)-th detector pixel centered at coordinates $\boldsymbol{x}_{i,j}$, then $C(i, j)$ is defined as:

$$C(i, j) \sim \mathscr{P}(A_{i,j} \cdot \mathrm{Dose}_{i,j} \cdot g(\boldsymbol{x}_{i,j})), \tag{1}$$

where $\mathscr{P}$ is Poisson process, $A_{i,j}$ is the area of the (i, j)-th pixel, and $\mathrm{Dose}_{i,j}$ is the incoming electron dose at the (i, j)-th pixel.

The overall gain, denoted by C_{gain}, measures the average number of digital counts that a single incident electron gives rise to. The detector response $R_{i,j}$ at (i, j)-th pixel is defined as:

$$R_{i,j} = C_{\mathrm{gain}} \cdot C(i, j). \tag{2}$$

The point spread function (PSF) of the detector specifies the blurring of each electron contribution when detected at a pixel and propagates it into the neighboring pixels. Hence, the data recorded at pixel (i, j)

denoted by $g^*(\boldsymbol{x}_{i,j})$, is obtained by performing convolution ($\otimes$). with the PSF of the detector:

$$g^*(\boldsymbol{x}_{i,j}) = R(i,j) \otimes \mathrm{PSF}_{\mathrm{detector}}(\boldsymbol{x}_{i,j}). \tag{3}$$

In reality, instead of using the $\mathrm{PSF}_{\mathrm{detector}}$, we often use the MTF, which is defined by the modulus of the $\mathrm{PSF}_{\mathrm{detector}}$. The reason is that the MTF is easier to measure on real systems than the $\mathrm{PSF}_{\mathrm{detector}}$ [9]. The MTF describes the attenuation of the sinusoidal image irradiance components as a function of spatial frequency. The MTF is commonly parametrized [45] as:

$$\mathrm{MTF}(\boldsymbol{\omega}) = \frac{a}{1+\alpha\boldsymbol{\omega}^2} + \frac{b}{1+\beta\boldsymbol{\omega}^2} + c, \tag{4}$$

where $\boldsymbol{\omega}$ is a two-dimensional spatial frequency vector; a, b, c, $\alpha > 0$, and $\beta > 0$ are the parameters. Those parameters depend on the type of detector and the electron dose used to capture the micrographs and are independent of the specimen.

System Our system contains two simulators as shown in Fig. 2, GPU Accelerated TEM (GPU-TEM) for rendering and Differentiable TEM (Diff-TEM) for solving inverse problems. **The schema of the simulator is shown in Fig. 9 in the supplementary materials**. The TEM simulator uses the multi-slice method [5] to compute the electron-specimen interaction, which is an imperative algorithm with complex control flows and is hard to be expressed in automatic differentiable operators of PyTorch [35]. Therefore, to speed up and differentiate this computation, we use Taichi [14, 15] and implement the calculation using Taichi kernels. However, we encountered limitations when using a combination of PyTorch and Taichi for simulating highly complex scenes due to memory access constraints. To overcome this challenge, we develop a non-differentiable GPU-accelerated simulator using CUDA.

The GPU-TEM receives the scene with the same configuration as used for acquisition with the real cryo-EM, and the optimized parameters (e.g., MTF) from the Diff-TEM as inputs. It then simulates the synthetic data that is similar to real-world data. The Diff-TEM receives real cryo-EM data and enables gradient computation, e.g., for estimating MTF parameters or denoising real data. By using this design, our system resolves two limitations of the TEM Simulator by Rullgård *et al.* [36]: (1) the need for user-estimated parameters, and (2) a time-consuming simulation process for scenes containing many instances of macromolecular structures.

Following the forward model, the simulator performs two tasks. First, assemble a model of the specimen and calculate the scattering potential of the specimen. This task is performed by a component that we call a phantom generator. Second, simulating the image formation process is further split into electron-specimen interaction, electron wave propagation, and intensity detection.

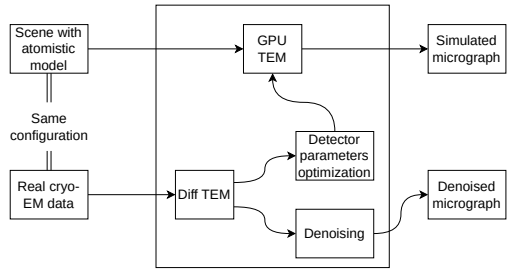

Figure 2: Our GPU-TEM simulator receives the scene configuration and detector parameters estimated from real data by our Diff-TEM simulator to generate a simulated micrograph. Our Diff-TEM simulator can also denoise real-world data, the results of which can then be used for tomographic reconstruction.

To accelerate the TEM simulator, we developed the GPU-TEM simulator that exploits parallelism of GPU hardware using the CUDA API [33]. A large portion of the simulator pipeline is implemented in CUDA, while there are some stages where we resort to CPU implementation.

In the phantom generator, the most time-consuming task is computing the scattering potential of particles. The scattering potential of one particle is calculated by the sum of the scattering potential of all atoms that form the particle. In the original TEM Simulator [36], this task is performed sequentially. To overcome this bottleneck in our GPU-TEM, we facilitate a CUDA kernel for computing the scattering potential for one atom, so the scattering potential of all atoms in the particle is computed in parallel at the atom level.

To calculate the electron-specimen interaction of each slice, we first determine which structures of the specimen are within the slice. Then we calculate the scattered electron wave after it interacts with these particles. This task is also performed sequentially in the original TEM Simulator [36]. We make another CUDA kernel in our GPU-TEM simulator to compute the scattered electron wave after it interacts with one structure, so the electron-specimen interaction of each slice is computed in parallel at the particle level. All CUDA kernels are called with 256 blocks, and each block contains 256 threads. These numbers were determined experimentally.

Our Diff-TEM simulator is differentiable, so all continuous parameters of different parts of the simulator can be optimized from the real data via backpropagation and gradient descent. We demonstrate this capability through two examples: (1) MTF parameter estimation and (2) denoising. From the pipeline, the final projection is obtained by adding Poisson noise to the noise-free projection and applying the MTF afterward. Therefore, to estimate the MTF, we perform backpropagation from real projections and use gradient descent for optimization. To denoise the real projections with gradient-based optimizations, we estimate the gradients of the Poisson process by using the method proposed by Schulman *et al.* [38] and apply gradient descent.

4 Optimization Components

The differentiability allows us to estimate parameters along the entire electron microscopy process pipeline. In this paper, we focus on detector parameters at the very end of the microscopy simulation.

4.1 MTF Parameters Estimation

Because MTF parameters are independent of the specimen, we can optimize these parameters by using a few cropped parts of the background (crops) from the real projections. These crops are $M \times N$ pixels in size. They are considered as the labelled projections $I_{l_1}, I_{l_2}, \ldots, I_{l_n}$. The predicted projections $I_{p_1}, I_{p_2}, \ldots, I_{p_n}$ are created using our simulator with the same configuration for the sample, electron beam, and optical system. The specimen of the predicted projection contains only the background. The predicted projections have the same size as the labeled projections. Labeled and predicted projections are normalized to the range $[0, 1]$. From the Equation 4, we can determine that optimizing MTF parameters is a regression problem. Applying the MTF is performed in the frequency domain, which motivates us to choose a loss function that can capture the differences between the predicted and the labeled projection in both the spatial and the frequency domains. Therefore, in this work, we choose focal frequency loss (FFL) [17] combined with mean square error (MSE) by summing them into one loss function. A brief description of the FFL loss and its γ parameter are presented in the Appx. C of the supplemental materials. The computational graph for estimating the MTF parameters is shown in Fig. 3.

The optimization can be formally described as:

$$\arg\min_{\Theta_D} \mathbb{E}_{\tilde{I}\sim\mathscr{P}(\boldsymbol{I}),I_l\sim\mathrm{ND}} \left[\mathscr{L}(\mathrm{MTF}_{\Theta_D}\left(\tilde{I}\right),\, I_l)\right], \tag{5}$$

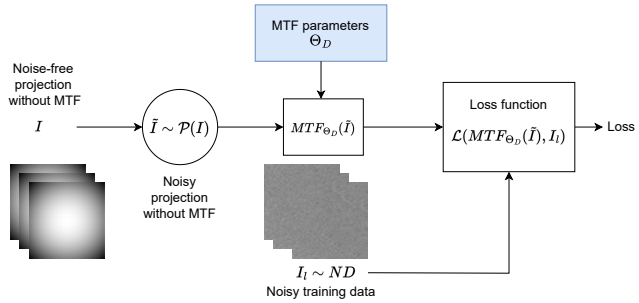

Figure 3: Compute Graph of Detector Parameter Estimation: The Diff-TEM first receives noisy training data, computes the loss with simulated noisy data, applies the predicted MTF, and performs gradient descent for optimizing MTF parameters.

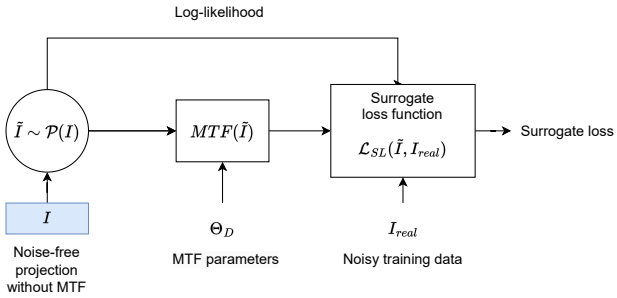

Figure 4: Compute graph for denoising: The projection without MTF is the optimization target. Log-likelihood values are used in our surrogate loss function along with the comparison loss that computes the difference between noisy projection predictions and noisy projections in a dataset.

where Θ_D is the set of MTF parameters, $\boldsymbol{I}$ a noise-free projection without MTF, ND is the noisy dataset, $\mathscr{L}$ is the loss function and $\mathscr{P}(\lambda)$ the Poisson distributions with its samples, the noisy projections $\tilde{I}$. ND can be the synthetic or real-world data. We conducted experiments for both datasets. We also conducted an experiment with different values γ of FFL. The MTF parameters are optimized using the Adam optimizer [18] ($\beta_1 = 0.8, \beta_2 = 0.992, \mathrm{lr} = 0.1$). We experimented with different options for $\mathscr{L}$ in Sect. 5.2 and found a combination of FFL and MSE to perform best. The details of our optimization algorithm are shown in Algorithm 1 in the supplementary material.

4.2 Denoising

The computation flow of the optimization of denoising is shown in Fig. 4. The optimization target $\boldsymbol{I}$ (i.e., NOISE-FREE PROJECTION WITHOUT MTF in Fig. 4) is initialized with random values. By applying Poisson Noise $\mathscr{P}$ with expected value $\boldsymbol{I}$, we create $\tilde{I}$ (i.e. NOISY PROJECTIONS WITHOUT MTF) to which the MTF is applied to create a NOISY PROJECTION PREDICTION (see Fig. 4) which is compared to the real noisy training data I_{real} in $\mathscr{L}_{\text{SL}}$. This step is also illustrated in Fig. 9 in the supplementary materials.

Since our simulator is differentiable, we can do inverse computations via gradient descent, which means we can re-formulate the denoising problem into an inverse problem. That is, given noisy data and a differentiable compute pipeline, to recover the noise-free projections without MTF, which are in the inputs to our forward computation. The optimization objective can be formulated as:

$$\arg\min_{\boldsymbol{I}} \mathbb{E}_{\tilde{I} \sim \mathscr{P}(\boldsymbol{I})} \left[\mathscr{L}_{\text{SL}} \left(\tilde{I},\, I_{\text{real}} \right) \right], \tag{6}$$

where $\boldsymbol{I}$ is the recovered denoised projection without MTF, $\mathscr{L}_{\text{SL}}$ the surrogate loss function devised from the work of stochastic compute graph [38] defined as Equation 7 and I_{real} the real data (e.g., noisy SARS-CoV-2 projections). To finally retrieve a noise-free prediction that respects the detector parameters, we simply apply the MTF directly to the noise-free projection $\boldsymbol{I}$. The surrogate loss

$$\mathscr{L}_{\text{SL}}(\tilde{I}, I_{\text{real}}) = \text{mean}\left(\mathscr{L}\left(I_{\text{real}} - \text{MTF}\left(\tilde{I}\right) \right) \cdot \ln p\left(\tilde{I} | \boldsymbol{I} \right) \right) \quad \text{with } \tilde{I} \sim \mathscr{P}(\boldsymbol{I}) \tag{7}$$

weighs the loss function $\mathscr{L}$ with the log-probabilities of the sampled noise to optimize to compensate gradient estimation for the Poisson noise. For denoising experiments, we choose MSE for $\mathscr{L}$.

The whole optimization algorithm is shown in Algorithm 2 in the supplementary material. In the denoising optimization, we also use Adam [18] as the optimizer with ($\beta_1 = 0.8, \beta_2 = 0.992, \mathrm{lr} = 1.0$).

Table 1: Performance evaluation using varying viral models in terms of number of atoms on top and using varying amounts of virus instances on the bottom. We compare the rendering times using the original simulator and our GPU-TEM.

VIRIONS	# ATOMS	TEM [36]	GPU-TEM (OURS)
TMV	12505	45s	**4s**
ZIKA	21149	1m 51s	**1m 46s**
SARS-CoV-2	$\approx$ 13 millions	7m 25s	**6m 4s**
VIRION	# INSTANCES	TEM [36]	GPU-TEM (OURS)
ZIKA	1	1m 51s	**1m 46s**
	10	**2m 16s**	3m 10s
	100	**2m 16s**	3m 48s
	1000	35m 27s	**6m 22s**
	5000	4h11m	**18m 9s**

5 RESULTS AND EVALUATION

5.1 Scalable TEM Simulation

Once we obtain the MTF parameters, we can generate synthetic micrographs with the same detector properties as those used for a reference real micrograph. For the purpose of fast rendering, we use the GPU-TEM implementation with several stages parallelized using CUDA.

We compare the performance of the original TEM simulator and our GPU-TEM. In the phantom generator, we use the SARS-CoV-2 virus model modeled using the modeling tool of Nguyen *et al.* [31] on the actual data[2] and the segmentation from the system of Nguyen *et al.* [30], ZIKA virus (Protein Data Bank (PDB) id 5IRE), and Tobacco Mosaic virus (TMV) (PDB id 2OM3). We demonstrate the scalability of our GPU-TEM in two aspects: (1) the number of atoms and (2) the number of virion instances. Relevant simulation parameters are presented in Table 3 in the supplementary materials, and a slice thickness of 50*nm* is used for all experiments. The performance of each simulator for a single instance of each type of virus is shown in Table 1. All performance experiments were done on the same workstation with one Intel(R) Xeon(R) CPU E5-2687W v4 3GHz, and one NVIDIA RTX 3090 GPU.

From the top of Table 1, it can be seen that for a single instance of TMV, and a single instance of SARS-CoV-2, our GPU-TEM is faster than the baseline [36]. The performance for a single instance of the ZIKA virus is similar for GPU-TEM and the baseline. To better understand the simulator performance, we measure the perfor-

[2] `https://www.ebi.ac.uk/emdb/EMD-33297`

mance on a varying number of ZIKA virion instances. The results are shown at the bottom of Table 1. Our GPU-TEM significantly outperforms the original implementation for scenes containing a high number of instances which are close to the real numbers of instances in real microscopy experiments. The original version of the baseline [36] could not simulate a single instance of SARS-CoV-2 because of memory demand exceeding the 256 GB of RAM available on our machine, so we adapted it by changing some structures and parameters in the original code of the baseline to simulate this scene for comparison. However, the adapted version can still not simulate more than two SARS-CoV-2 virions, while our GPU-TEM can simulate a scene with more than 20 instances of SARS-CoV-2. Further parallelization of the simulator pipeline stages, such as the multi-slice stage, would result in an even larger performance gap. With the proposed approach, we can overcome the time-consuming bottleneck for the simulation of complex scenes with a large number of particle instances. When the number of instances is high, two GPU kernels described in Equation 3 perform better than a CPU version. When the number of instances is low, the GPU-TEM is slower than the baseline, due to the data transfer many times between the CPU and the GPU.

To illustrate the capability of our GPU-TEM simulator, in Fig. 5, we also show the result of simulating a larger scene with multiple SARS-CoV-2 virions side-by-side with the real micrographs from Yao *et al.* [44]. The presented specimens are on the same level of complexity with respect to the number of atoms/particles. In Fig. 5, we do not simulate the gold beads since they are used for alignment and do not affect the biological structures in the specimen. At first glance, the signal of virions in the simulated image and the real image are similar. The background noise of our simulated image is darker than the real image because the interaction between the water and electron beam is not accurate enough. We plan to improve this stage in the future. The real data is also preprocessed by applying contrast transfer function (CTF) correction using Novactf [40, 44] and we do not have the parameters that the owner of the dataset used for CTF correction so we could not apply for our simulated micrograph. As the result, the contrast of the real image and simulated image are not similar. In simulated images, there are some bright spots and dark spots in the inner part of virions. These spots are the results of the inaccuracy of atomistic models. The inner parts of the SARS-CoV-2 virion in our atomistic models are modeled based on the knowledge from the work of Yao *et al.* [44].

5.2 Reverse-Engineering MTF

For estimating MTF parameters, we perform experiments on both synthetic and real-world data. The synthetic data is created from the envelope of SARS-CoV-2 and TMV virions. The real-world dataset is provided by Yao *et al.* [44] and consists of 20 tilt series containing several SARS-CoV-2 virions each. For the purpose of parameter optimization, we only use cropped sub-projections containing background. The real data was acquired using a Titan Krios microscope with K3 detector from Gatan, operated at a voltage of 300 keV. The ground truth of MTF parameters for real data was obtained by performing curve-fitting the MTF curve of Gatan K3 detector[3]. The ground truth values of MTF parameters for the synthetic data TMV are provided by Rullgård *et al.* [36] in their simulation and calibrated with their real data which is a micrograph of TMV virions measured by a Philips CM 200 FEG TEM microscope. To demonstrate the generalization of the estimation method, we also experimented with synthetic data using varying MTF from the MTF previously determined using the synthetic data.

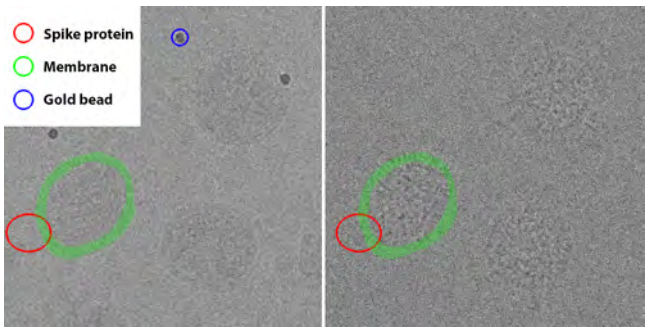

Figure 5: Real (left) and simulated (right) micrograph of a scene with multiple SARS-CoV-2 virions. We annotate some parts of one virion (spike protein and membrane) for comparison.

We conducted experiments for each loss function (MSE, MSE + FFL) with a varying number of background images of synthetic data to see the correlation between the performance of each loss and the numbers of projections. The number that performs best on synthetic data is used for the number of projections for experiments with real data. Furthermore, we carried out experiments with varying FFL loss γ values: $(0.0, 0.5, 1.0, 2.0)$. **The results of these experiments are shown in Fig. 10 in the supplementary materials**.

The effect of the MTF is difficult to visually inspect in the noisy cryo-EM projections because of the low SNR. Therefore, we applied both the ground truth MTF and the prediction MTF on the noise-free projections of the synthetic scene for evaluation. The details of the experimental settings for the synthetic data, i.e., the number of background projections, and the ground truth values of the MTF parameters (a, b, c, α, β) are shown in Table 4 in the supplementary materials.

To evaluate the effect of the MTF on the noise-free projection, we calculated the contribution of each frequency for the noise-free projection and determined which frequency band yields the largest contribution by summing the magnitude for each band. The frequency bands (in units $\frac{1}{\text{pixel}}$) we used are $\{[0.0, 0.02), [0.02, 0.04), \ldots, [0.48, 0.5)\}$. Due to the low SNR, the information from high frequencies is not preserved in the projections. So the effects of the MTF are mostly presented in the low frequencies or the most contributing frequency band. If the predicted MTF aligns well with the most contributing frequency bands of the ground truth MTF, their effects on the noise-free projections are similar. Additionally, we compare the predicted and ground truth MTF by comparing the resulting images after applying each, the predicted and ground truth MTF on noise-free projections. We compare the resulting images by reporting the mean and maximum pixel-wise difference. If two different MTF predictions align with the ground truth MTF to the same degree, we choose the MTF, which has the smaller mean and maximum difference values. Moreover, while MTF parameters are defined for the whole image, different sets of MTF parameters can result in a similar output. Therefore, directly comparing the estimated MTF parameters with the target parameters is not meaningful. This is why instead of comparing the parameters directly, we compare the resulting images created using estimated and target MTF parameters.

The correlation between the number of background projections and the performance of each loss function on synthetic data is shown in Fig. 10 (a, b) in the supplementary material. The MSE performance decreases when the number of background projections is greater than 10, the gap between the predicted MTF and the ground truth MTF increases, and the mean and maximum differences also increase. With 50 background projections, the results using the predicted MTF are blurry, and it is hard to see the details of individual TMV virions. On the contrary, the FFL ($\gamma = 0.5$) performance improves when increasing the number of background projections. The predicted MTF with five background projections is more aligned with the ground truth MTF than 10 or 20 background projections.

[3] https://www.gatan.com/K3#resources

However, the estimation from 50 background projections is better than five background projections. The results using the predicted MTF based on 50 projections have the smallest mean and maximum difference values. The comparison of the best MSE and FFL results is shown in Fig. 10 (a, b) in the supplementary material. It is clear that FFL is a better choice for optimization than MSE.

We also investigated how the change of FFL γ parameter affects the MTF estimation for different settings with 50 background projections. The results are shown in Fig. 10 (c, d) in the supplementary material. The best predicted MTF for SARS-CoV-2, TMV, and synthetic setting are obtained by using $\gamma = 0.5$, $\gamma = 2.0$, $\gamma = 2.0$, for FFL respectively. The best γ value for each experiment is different because the settings (type of detector, electron dose, etc.) of different experiments are not the same. The distribution of frequencies in background projections is different. The results show that our system works effectively for different types of MTF curves. The first type is the real MTF (TMV and SARS-CoV-2), which gives high modulation in the most contributing frequency bands. This results in the preservation of virion details. The second type is the synthetic MTF, where the modulation decreases in the most contributing frequency band, resulting in blurred images with applied MTF.

In summary, our predicted MTF for each case is aligned well with the ground truth in the most contributing frequency band and produces a visually similar effect on the noise-free projections.

5.3 Denoising Micrographs

Our differentiable simulator Diff-TEM allows us to predict the noise-free version of our micrographs. We have investigated this capability by means of multiple experiments, whereby we compare the denoising capabilities of our approach in synthetic and real settings qualitatively and quantitatively.

First, we conducted experiments using a synthetic dataset generated with our simulator for TMV virions. In this setting, we obtain clean ground-truth data that are synthesized by bypassing the Poisson. Note that in simulation and the real electron microscopy process, the dose, measured in electrons per nm^2 per projection, is a major factor determining SNR. We tested our denoising method with synthetic datasets with varying doses. The higher the dose is, the higher SNR is. To generate a series of 60 projections, 100 electrons/nm^2 per projection is a typical low dose, while 2000 electrons/nm^2 is a typical high dose. We tested datasets acquired using doses of $\{100, 200, 300, 500, 700, 1000, 1500, 2000\}$ per nm^2 per projection. We used such doses because it is possible to use a very high dose (e.g., 1000) if the number of total projections is low; thus, the total dose stays the same. Moreover, the number of available projections of the same scene (i.e., same tilt angle and specimen) also affects the denoising quality. We tested our method on the numbers of available projections of $\{2, 5, 10, 25, 50\}$. We denote varying data settings with pairs, i.e., (dose, number of available projections).

Moreover, to compare with the original Noise2Noise model [24] and Topaz-Denoise [1], in the synthetic setting, we trained a Noise2Noise model and a Topaz-Denoise model (denoted as `Topaz T`) with our synthetic data (number of projections are 4500 or 5000 with electron dose of 100 electrons/nm^2). We also use the pre-trained Topaz-Denoise model (denoted as `Topaz P`) for comparison. Note that in the synthetic setting, we do not apply any data augmentation in our method.

Using the synthetic data with ground truth in the setting (100, 2) is shown in Fig. 11 in the supplementary materials. In Fig. 11 (a), we have a very limited amount of projections (i.e., only 2) and a low dose leading to a low SNR. In settings like this, where data are scarce and full of noise, a natural choice is to average the two available projections. Although the denoised result is not close to the noise-free reference, our method outperforms the simple average of the two available projections in terms of structural similarity

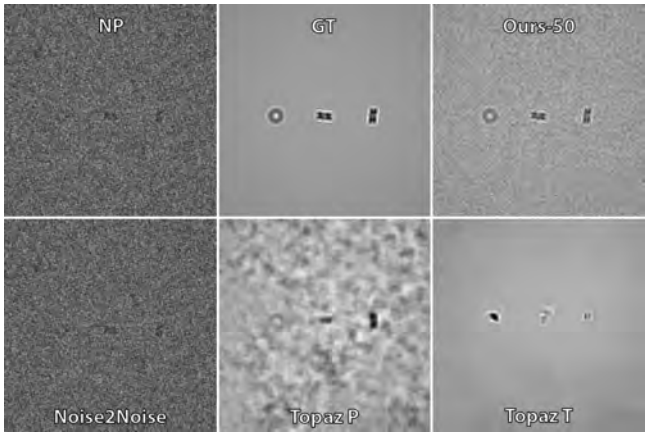

Figure 6: Comparison of denoising the synthetic data: images in the top row show original noisy projection (NP) and clean ground truth projection (GT) and our denoising result, in the bottom row are denoising results using the trained Noise2Noise result (Noise2Noise), pre-trained Topaz-Denoise (`Topaz P`), and Topaz-Denoise trained on our noisy projections (`Topaz T`).

(SSIM) [42], MSE, peak signal-to-noise ratio (PSNR) and FFL. In the very high dose cases (i.e., 700, 1000, 1500, 2000), we show consistently better results than simple averaging in terms of FFL. For example, the quantitative comparison in Fig. 11 (b) shows that the FFL of our method is roughly 4 times better than simple averaging. The quantitative comparisons of these two figures show that our physically-based Noise2Noise-like model is effective.

To compare with the original Noise2Noise [24] model and Topaz-Denoise [1], we present our results from the setting (100, 50) denoted as Ours-50 in Fig. 6. The figure also shows the results of a trained Noise2Noise model, `Topaz P` and `Topaz T`. The Noise2Noise model does not learn to denoise even if we have a large amount of available data (i.e., 5000). Therefore, we further tested the performance of the Topaz model with our synthetic dataset. The results show that the pre-trained model (i.e., `Topaz P`) generalizes well by filtering out high-frequency noise and preserving the particles, but it is worse than our method because the boundaries of particles are blurred. Our trained Topaz model (i.e., `Topaz T`) filters out most of the noise and recognizes the positions of the particles. However, it does not preserve the particles' shape, making it unusable. On the contrary, our method can work in the setting with less data (i.e., (100, 50)), which shows that our method is advantageous over training a Noise2Noise model [24] and a Topaz-Denoise model [1] from scratch.

It is noteworthy that the Noise2Noise model and `Topaz T` are trained with data settings (100, 5000) and (100, 4500), respectively. It is neither realistic nor viable to obtain such large datasets in real-world scenarios, which means these two models are unusable. However, `Topaz P` generalizes well in our setting, so we use it for data augmentation in the real-world scenario to generate more data for our denoising method. For more results of our method under synthetic settings, we refer the reader to the Appx. G of the supplementary materials.

In the experiments using real micrographs, we use the tilt series from Yao *et al.* [44], which show SARS-CoV-2 virions. The dose in the dataset is 320 electrons per nm^2 per projection. Therefore, without data augmentations, the data setting is (320, 1). As discussed in Section 2, our method is essentially a physical-model-based Noise2Noise [24], which means it cannot be trained with only one projection, so in our experiments, we apply a number of data augmentations. In the setting `Ours(T)`, the data augmentation with `Topaz P` is used, yielding one more projection, which means the data setting is (320, 2). In the setting denoted as `Ours(N)`, neigh-

Table 2: The comparison of SSIM, PSNR, and FFL metrics for different settings using the input data `Noisy`. `Ours(T+N)` results for three metrics are shown in bold. All these values are between `Noisy` and `Topaz`'s results. `Topaz` results contain the least noise but lose the details of structures, while the details are shown better in `Noisy` but surrounded by noise. The results in the table show that our method with the setting `Ours(T+N)` is a compromise for both noise reduction and details preservation.

	SSIM ↑					PSNR ↑					FFL ↓				
	Noisy	Topaz	Ours(T)	Ours(N)	Ours(T+N)	Noisy	Topaz	Ours(T)	Ours(N)	Ours(T+N)	Noisy	Topaz	Ours(T)	Ours(N)	Ours(T+N)
Noisy	-	0.6017	0.4139	0.7395	**0.6547**	-	65.8250	65.8270	70.1500	**66.1000**	-	0.0096	0.0082	0.0032	**0.0095**
Topaz	0.6017	-	0.6894	0.7184	**0.8651**	65.8250	-	72.2000	68.9700	**75.0400**	0.0096	-	0.0012	0.0045	**0.0008**
Ours(T)	0.4139	0.6894	-	0.7797	0.8001	65.8270	72.2000	-	71.6200	76.2800	0.0082	0.0012	-	0.0020	0.0001
Ours(N)	0.7395	0.7184	0.7797	-	0.9098	70.1500	68.9700	71.6200	-	71.6700	0.0032	0.0045	0.0020	-	0.0029
Ours(T+N)	0.6547	0.8651	0.8001	0.9098	-	66.1000	75.0400	76.2800	71.6700	-	0.0095	0.0008	0.0001	0.0029	-

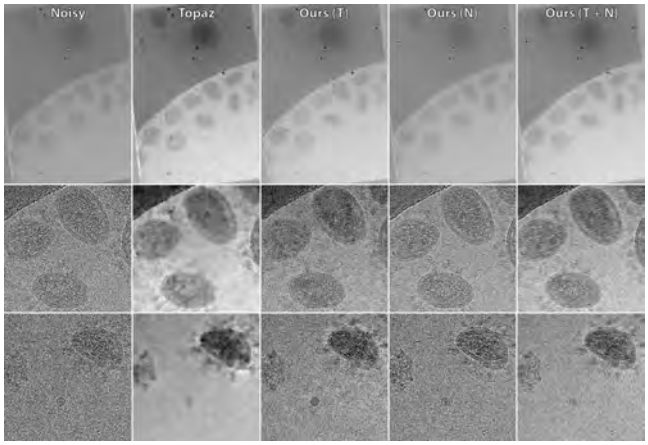

Figure 7: `Noisy`: the real noisy projection; `Topaz`: the real projection denoised by `Topaz P`; Top: Full views; Middle: Zoom-in of the first row. In the bottom row is another example of our denoising method, `Ours(T+N)` contains less noise while preserving boundaries.

boring projections of a projection are aligned and used, so the data setting is (320, 3). In the setting `Ours(T+N)`, `Topaz P` and aligned neighboring projections are used, so the data setting is (320, 6). The case of `Ours(T+N)` matches exactly our Algorithm 2 in the supplementary material, while in the case of `Ours(T)`, $P_n = \{P\}$, and in the case of `Ours(N)`, preprocessing with Topaz-Denoise is not used (i.e., $P_{\text{train}} = \text{Normalize}(P_n)$). In the real setting, we do not have noise-free ground truth projections to compare against, so we first present the results of the qualitative comparison.

On the top of Fig. 7, we can see `Topaz` has block-like artifacts and `Noisy` contains more noise while others have minor visual differences. However, if we zoom in on the images, as shown in the bottom of Fig. 7, we can see (1) `Topaz` has more blurring artifacts, (2) `Noisy` has the most noise, and (3) `Ours(T+N)` is better than `Ours(N)` and `Ours(T)` in that it has less high-frequency noise while preserving clear boundaries. This makes our method `Ours(T+N)` the best among these five from a qualitative point of view because of its good balance between denoising and preserving boundaries.

We also present quantitative results in Table 2. It is important to note that we could not have the ground truth for denoising experiments with the real dataset. That is why we compare quantitatively with `Noisy` and `Topaz` results. Based on the bottom images from Fig. 7, we can conclude that `Topaz` contains the least noise while `Noisy` contains the most noise. With this conclusion, the values in bold in the table show that (1) `Ours(T+N)` is the closest one to `Topaz` - the SSIM, PSNR, FFL to `Topaz` are better than `Ours(T)` and `Ours(N)`; (2) `Ours(T+N)` has a moderate amount of noise compared to `Ours(T)` and `Ours(N)` - the SSIM, PSNR to `Noisy` is lower than `Ours(N)`, but higher than `Ours(T)`. This means `Ours(T+N)` is a good compromise between removing noise and preserving details, echoing our qualitative comparison.

In addition to the example on the top two rows of Fig. 7, we give another example of denoising real data. We can draw similar conclusions from the last row in Fig. 7. `Topaz` has the least noise but compromised boundaries, while `Ours(T+N)` has a moderate amount of noise but also clearer boundaries and more visibility of spikes. Denoising experiments for both synthetic and real data were conducted on a Linux system with two NVIDIA Quadro RTX 8000 GPUs. The experiments took around three to five minutes to obtain reasonable results. The processing time also depends on the chosen number of iterations, as with any other procedures involving iterative optimizations.

The applications of such denoising are also useful for tomographic reconstruction. In the left part of Fig. 8, we show the mid-slice from the volumes reconstructed using the SIRT reconstruction technique from IMOD [22] from original noisy projections (left), from denoised projections using `Topaz` (middle), and from denoised projections using our approach `Ours(T+N)` (right). One can clearly see that `Topaz` and our approach are better than reconstructions from the original noisy projections. Moreover, our approach retains more details that are smudged out by `Topaz`.

Additionally, in the right part of Fig. 8, we show DVR rendering of the same volumes using the same rendering parameters apart from the threshold, which was manually adapted for each case. One can see that `Topaz` cleans the volume extensively but removes parts of the virions and high-frequency details from the structures. This is not the case for results using our approach. Although more noise is present in the volume, the high-frequency details are retained. Both approaches are clearly a better choice than rendering the original noisy volume.

6 Discussion

A common problem in biological research, especially in TEM, is the need for ground-truth data. This poses a significant challenge when developing and evaluating image processing and analysis techniques. However, our system provides a promising solution to overcome this limitation, as demonstrated by the comparable signal levels of the showcased virions with actual SARS-CoV-2 data. Additionally, it is essential to accurately estimate the parameters of the detector used in the imaging process to generate augmented data that closely resembles real datasets. Our MTF parameters estimation experiments illustrate that our simulator effectively addresses this requirement. In addition to addressing the lack of ground-truth data, our simulator can tackle denoising tasks even in scenarios with a limited amount of real data. Comparing our approach with other existing methods, such as Topaz-Denoise [1], we have demonstrated that our simulator excels in preserving fine structural details of the specimen.

We discussed the results of our work with an independent domain expert. The expert is a physicist specializing in biophysics. He has 20 years of experience in EM specifically. He is a senior staff scientist in the electron microscopy laboratory at our university and usually works with EM data.

We showed him the MTF parameters that we estimate from the real TEM micrographs (Fig. 10). He pointed out that nowadays, the

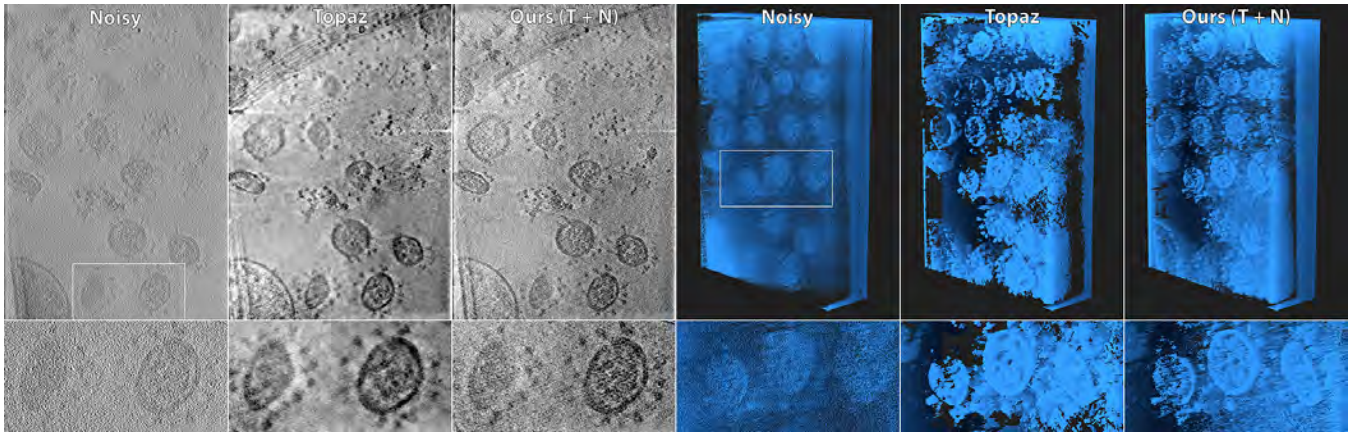

Figure 8: Tomographic reconstructions from the tilt series (left) and DVR renderings (right).

MTF parameters are not estimated by the end user but are predetermined by the manufacturer. However, our estimation would be very useful for electron microscope manufacturers since this process is usually quite tedious. He also noted that it would benefit users if we expanded our system to estimate the CTF parameters.

We also showed him our denoising results Fig. 6, Fig. 7, Fig. 8, he pointed out that the usefulness of the denoising strongly depends on the use case. He said that if a user wants to see the structures clearer and does not care about their distortion, then Topaz's results are better. However, if the user is interested in the structure details and their orientation, then our results are better. Both denoising approaches have their benefits and disadvantages. Moreover, he confirmed that our denoising approach retains more information than the Topaz approach. He also advised us to evaluate our approach using our denoised projections for high-resolution reconstruction or tomographic reconstruction and comparing the particle orientation correctness. We plan to do these comparisons in the future.

6.1 Limitations and Future Work

This study demonstrated the potential of differentiability within certain components of the cryo-EM pipeline. However, analogous to most other simulators, our emphasis was predominantly on modeling the specimen and not on the cryogenic water of the background. The electron-specimen interaction is elementary, where interaction between electrons and the background, and inelastic effects are deliberately omitted. Therefore, enhancing the interaction modeling is necessary to get more similar-to-indistinguishable micrographs. Because the calculation for all slices of the specimen is still performed on the CPU, our GPU-TEM still takes time to transfer the data from GPU to CPU. In the future, we plan to perform this calculation in parallel and apply wave propagation in optics for the electron wave, then use the differentiable capacity of the wave propagation method to perform tomography reconstruction or estimate parameters estimation of the sample and the electron source. Some hyperparameters for MTF estimation, such as γ for FFL and the initial values for detector parameters, are selected randomly. In the future, we plan to develop a method for determining the distribution of frequencies, then we can guess which frequency components are hard to synthesize and choose a good γ for FFL from the distribution of frequencies. Moreover, estimation for CTF parameters is beneficial for users, and we also plan to optimize these parameters as well. Our denoising method is a physically-based Noise2Noise model, which shares the same limitation as in the original Noise2Noise [24]. This means that our method needs multiple projections to denoise one projection, while in real-world scenarios, the data are limited. To mitigate this limitation, we believe it is possible to use neural networks to learn and transfer the patterns from various data. Our simulator works well for direct detectors, so our current noise model may not be accurate enough. This issue may lead to some biases in our denoising method, so an expansion for other types of detectors that introduce other types of noise is also in our future endeavors. With the differentiability of our simulator, future work can be done to embed a neural network into the physical model of our simulator, taking advantage of the generalizability of neural networks while preserving physical intuitions.

7 Conclusion

With this work, we presented a system combining GPU-TEM and Diff-TEM simulators that enable rendering large micrographs *on par* with the physical dimensions and complexity of real-world microscopy. Nonetheless, our system not only performs forward calculation but also enables backward computation, which gives rise to a new way to solve inverse problems in microscopy imaging. We demonstrate its ability by showcasing two examples, detector parameter estimation and denoising, which are typical inverse problems.

Our GPU-TEM simulator outperforms the baseline TEM simulator when simulating a scene containing a high number of complex structures. Our method for detector parameters estimation learns from the real data, automating the time-consuming manual calibration. Moreover, users can use this functionality to guess a reasonable set of parameters and fine-tune them manually if desired. This narrows the search space and speeds up tuning. Our method of denoising shows better results on our synthetic datasets than existing DL-based methods. On real micrographs, with data augmentation using a state-of-the-art denoiser and neighboring projections, we show improvements compared to the state-of-the-art. For downstream applications, our denoised projections can contribute to better tomographic reconstruction and thus improve the quality of visualizations.

Acknowledgments

The authors would like to thank Sai Li and his team at Tsinghua University, China for sharing the SARS-CoV-2 cryo-EM data for this work. This research has been funded by KAUST baseline funding BAS/1/1680-01-01 and KAUST CRG grant URF/1/4345-01-01.

References

[1] T. Bepler, K. Kelley, A. J. Noble, and B. Berger. Topaz-Denoise: general deep denoising models for cryoEM and cryoET. *Nature Communications*, 11(1):5208, Oct 2020. doi: 10.1038/s41467-020-18952-1

[2] H. Berman, K. Henrick, and H. Nakamura. Announcing the worldwide protein data bank. *Nature structural biology*, 10(12):980, 2003. doi: 10.1038/nsb1203-980

[3] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[4] T.-O. Buchholz, M. Jordan, G. Pigino, and F. Jug. Cryo-care: Content-aware image restoration for cryo-transmission electron microscopy

data. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pp. 502–506, 2019. doi: 10.1109/ISBI.2019.8759519

[5] J. M. Cowley and A. F. Moodie. The scattering of electrons by atoms and crystals. i. a new theoretical approach. *Acta Crystallographica*, 10(10):609–619, 1957.

[6] J. J. Donatelli and J. C. Spence. Inversion of many-beam bragg intensities for phasing by iterated projections: Removal of multiple scattering artifacts from diffraction data. *Physical review letters*, 125(6):065502, 2020.

[7] D. Fanelli and O. Öktem. Electron tomography: a short overview with an emphasis on the absorption potential model for the forward problem. *Inverse Problems*, 24(1):013001, 2008.

[8] A. S. Frangakis and R. Hegerl. Noise Reduction in Electron Tomographic Reconstructions Using Nonlinear Anisotropic Diffusion. *Journal of Structural Biology*, 135(3):239–250, 2001. doi: 10.1006/jsbi.2001.4406

[9] J. W. Goodman. *Introduction to Fourier optics*. Roberts and Company Publishers, 3rd. ed., 2005.

[10] H. Gupta, M. T. McCann, L. Donati, and M. Unser. CryoGAN: A New Reconstruction Paradigm for Single-Particle Cryo-EM Via Deep Adversarial Learning. *IEEE Transactions on Computational Imaging*, 7:759–774, 2021. doi: 10.1109/TCI.2021.3096491

[11] A. Hamdi, S. Giancola, and B. Ghanem. MVTN: Multi-View Transformation Network for 3D Shape Recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–11, October 2021.

[12] J. Hasselgren, J. Munkberg, M. Salvi, A. Patney, and A. Lefohn. Neural Temporal Adaptive Sampling and Denoising. *Computer Graphics Forum*, 2020. doi: 10.1111/cgf.13919

[13] B. Himes and N. Grigorieff. Cryo-tem simulations of amorphous radiation-sensitive samples using multislice wave propagation. *IUCrJ*, 8(6), 2021.

[14] Y. Hu, L. Anderson, T.-M. Li, Q. Sun, N. Carr, J. Ragan-Kelley, and F. Durand. DiffTaichi: Differentiable Programming for Physical Simulation. *ICLR*, 2020.

[15] Y. Hu, T.-M. Li, L. Anderson, J. Ragan-Kelley, and F. Durand. Taichi: a language for high-performance computation on spatially sparse data structures. *ACM Transactions on Graphics (TOG)*, 38(6):201, 2019.

[16] X. Huang, S. Li, and S. Gao. Progress in filters for denoising cryo-electron microscopy images. *Journal of Peking University. Health sciences*, 53(2):425–433, March 2021. doi: 10.19723/j.issn.1671-167x.2021.02.033

[17] L. Jiang, B. Dai, W. Wu, and C. C. Loy. Focal frequency loss for image reconstruction and synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13919–13929, 2021.

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] A. I. Kirkland and R. R. Meyer. "indirect" high-resolution transmission electron microscopy: Aberration measurement and wavefunction reconstruction. *Microscopy and Microanalysis*, 10(4):401–413, 2004.

[20] E. J. Kirkland. *Advanced Computing in Electron Microscopy*. Springer, 3rd. ed., 2020.

[21] H. Kniesel, T. Ropinski, T. Bergner, K. S. Devan, C. Read, P. Walther, T. Ritschel, and P. Hermosilla. Clean implicit 3d structure from noisy 2d stem images. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2022.

[22] J. R. Kremer, D. N. Mastronarde, and J. R. McIntosh. Computer visualization of three-dimensional image data using imod. *Journal of structural biology*, 116(1):71–76, 1996.

[23] W. Kühlbrandt. The resolution revolution. *Science*, 343(6178):1443–1444, 2014.

[24] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila. Noise2Noise: Learning Image Restoration without Clean Data. In J. Dy and A. Krause, eds., *Proceedings of the 35th International Conference on Machine Learning*, vol. 80 of *Proceedings of Machine Learning Research*, pp. 2965–2974. PMLR, 10–15 Jul 2018.

[25] B. D. Levin. Direct detectors and their applications in electron microscopy for materials science. *Journal of Physics: Materials*, 4(4):042005, 2021.

[26] H. Li, H. Zhang, X. Wan, Z. Yang, C. Li, J. Li, R. Han, P. Zhu, and F. Zhang. Noise-Transfer2Clean: denoising cryo-EM images based on noise modeling and transfer. *Bioinformatics*, 02 2022. doi: 10.1093/bioinformatics/btac052

[27] T.-M. Li, M. Aittala, F. Durand, and J. Lehtinen. Differentiable Monte Carlo Ray Tracing through Edge Sampling. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 37(6):222:1–222:11, 2018.

[28] I. Lobato and D. Van Dyck. Multem: A new multislice program to perform accurate and fast electron diffraction and imaging simulations using graphics processing units with cuda. *Ultramicroscopy*, 156:9–17, 2015.

[29] X. Mao, C. Shen, and Y.-B. Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. *Advances in neural information processing systems*, 29, 2016.

[30] N. Nguyen, C. Bohak, D. Engel, P. Mindek, O. Strnad, P. Wonka, S. Li, T. Ropinski, and I. Viola. Finding nano-ötzi: Cryo-electron tomography visualization guided by learned segmentation. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–18, 2022. doi: 10.1109/TVCG.2022.3186146

[31] N. Nguyen, O. Strnad, T. Klein, D. Luo, R. Alharbi, P. Wonka, M. Maritan, P. Mindek, L. Autin, D. S. Goodsell, et al. Modeling in the time of covid-19: Statistical and rule-based mesoscale models. *IEEE transactions on visualization and computer graphics*, 27(2):722, 2021.

[32] E. Nogales and S. H. Scheres. Cryo-em: a unique tool for the visualization of macromolecular complexity. *Molecular cell*, 58(4):677–689, 2015.

[33] NVIDIA, P. Vingelmann, and F. H. Fitzek. Cuda, release: 10.2.89, 2020.

[34] C. Ophus and T. Ewalds. Guidelines for quantitative reconstruction of complex exit waves in hrtem. *Ultramicroscopy*, 113:88–95, 2012.

[35] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[36] H. Rullgård, L.-G. Öfverstedt, S. Masich, B. Daneholt, and O. Öktem. Simulation of transmission electron microscope images of biological specimens. *Journal of microscopy*, 243(3):234–256, 2011.

[37] W. Schottky. Über spontane stromschwankungen in verschiedenen elektrizitätsleitern. *Annalen der physik*, 362(23):541–567, 1918.

[38] J. Schulman, N. Heess, T. Weber, and P. Abbeel. Gradient Estimation Using Stochastic Computation Graphs. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pp. 3528–3536. MIT Press, Cambridge, MA, USA, 2015.

[39] M. Turk and W. Baumeister. The promise and the challenges of cryo-electron tomography. *FEBS Letters*, 594(20):3243–3261, 2020. doi: 10.1002/1873-3468.13948

[40] B. Turoňová, F. K. Schur, W. Wan, and J. A. Briggs. Efficient 3d-ctf correction for cryo-electron tomography using novactf improves subtomogram averaging resolution to 3.4 å. *Journal of structural biology*, 199(3):187–195, 2017.

[41] M. Vulović, R. B. Ravelli, L. J. van Vliet, A. J. Koster, I. Lazić, U. Lücken, H. Rullgård, O. Öktem, and B. Rieger. Image formation modeling in cryo-electron microscopy. *Journal of structural biology*, 183(1):19–32, 2013.

[42] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861

[43] S. Weiss and R. Westermann. Differentiable direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):562–572, 2022. doi: 10.1109/TVCG.2021.3114769

[44] H. Yao, Y. Song, Y. Chen, N. Wu, J. Xu, C. Sun, J. Zhang, T. Weng, Z. Zhang, Z. Wu, L. Cheng, D. Shi, X. Lu, J. Lei, M. Crispin, Y. Shi, L. Li, and S. Li. Molecular Architecture of the SARS-CoV-2 Virus. *Cell*, 183(3):730–738, 2020. doi: 10.1016/j.cell.2020.09.018

[45] J. Zuo. Electron detection characteristics of a slow-scan ccd camera, imaging plates and film, and electron image restoration. *Microscopy research and technique*, 49(3):245–268, 2000.