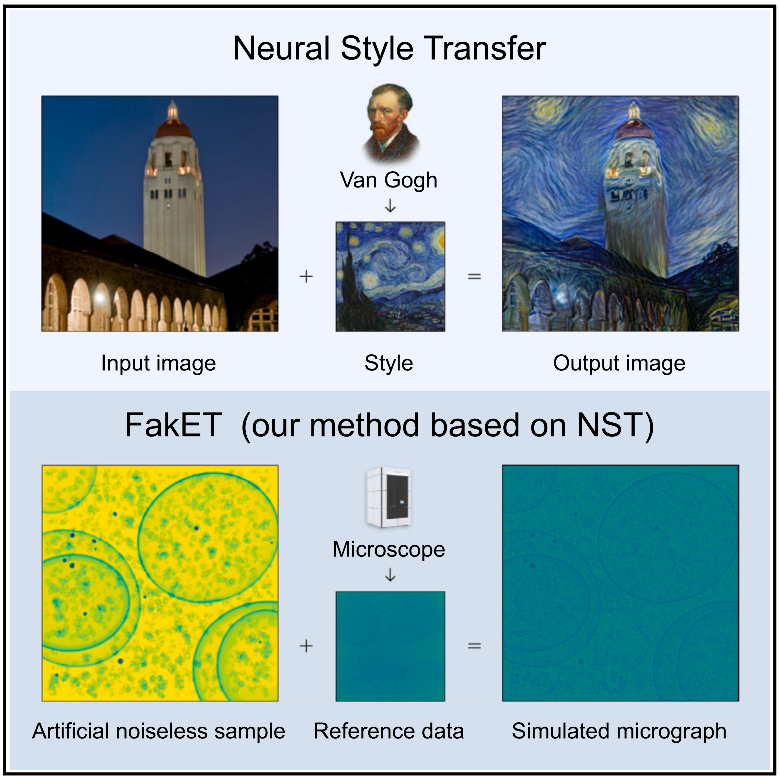
Structure

Resource

FakET: Simulating cryo-electron tomograms with neural style transfer

Graphical abstract



Highlights

- FakET adapted neural style transfer technique to simulate cryo-electron micrographs
- It produces high-quality data resembling the unlabeled reference micrographs
- The simulation cost is far lower than that of complex physics-based simulators
- Offers a practical choice of simulating fully labeled training data for deep learning

Authors

Pavol Harar, Lukas Herrmann,
Philipp Grohs, David Haselbach

Correspondence

pavol.harar@ista.ac.at (P.H.),
david.haselbach@imp.ac.at (D.H.)

In brief

Harar et al. demonstrate that neural style transfer, a technique traditionally employed for manipulating natural images, can be successfully adapted to simulate cryo-electron micrographs or tilt-series. This application provides the means to generate high-quality, fully labeled data for training deep neural networks, enhancing their efficacy on downstream tasks within computational microscopy.

Harar et al., 2025, Structure *33*, 820–827

https://doi.org/10.1016/j.str.2025.01.020

CellPress

CellPress
OPEN ACCESS

Structure

Resource

FakET: Simulating cryo-electron tomograms with neural style transfer

Pavol Harar,[1,2,3,4,5,7,*] **Lukas Herrmann,**[6] **Philipp Grohs,**[1,3,6] **and David Haselbach**[2,*]
[1]Mathematical Data Science (MDS), Faculty of Mathematics, University of Vienna, Vienna, Austria
[2]Haselbach Lab, Research Institute of Molecular Pathology (IMP), Vienna, Austria
[3]Research Network Data Science, University of Vienna, Vienna, Austria
[4]Department of Telecommunications, Faculty of Electrical Engineering and Communication, Brno University of Technology, Brno, Czech Republic
[5]Institute of Science and Technology Austria (ISTA), Klosterneuburg, Austria
[6]Johann Radon Institute for Computational and Applied Mathematics, Austrian Academy of Sciences, Linz, Austria
[7]Lead contact
*Correspondence: pavol.harar@ista.ac.at (P.H.), david.haselbach@imp.ac.at (D.H.)
https://doi.org/10.1016/j.str.2025.01.020

SUMMARY

In cryo-electron microscopy, accurate particle localization and classification are imperative. Recent deep learning solutions, though successful, require extensive training datasets. The protracted generation time of physics-based models, often employed to produce these datasets, limits their broad applicability. We introduce FakET, a method based on neural style transfer, capable of simulating the forward operator of any cryo transmission electron microscope. It can be used to adapt a synthetic training dataset according to reference data producing high-quality simulated micrographs or tilt-series. To assess the quality of our generated data, we used it to train a state-of-the-art localization and classification architecture and compared its performance with a counterpart trained on benchmark data. Remarkably, our technique matches the performance, boosts data generation speed 750 × , uses 33× less memory, and scales well to typical transmission electron microscope detector sizes. It leverages GPU acceleration and parallel processing. The source code is available at https://github.com/paloha/faket/.

INTRODUCTION

Recent developments in cryo-electron tomography (cryoET) allow to obtain high resolution representations of macromolecular complexes in their native cellular environment showing molecular interactions that are hardly accessible with other methods.[1] In cryoET, the imaged sample is in most cases a 100–200 nm thick slice of a frozen cell. From this slice, projection images are taken in a transmission electron microscope (TEM) from different rotation (tilt) angles. An artifact free reconstruction would require measurements using tilt angles that would complete the half circle. However, this is not feasible, due to limitations of the specimen holder, and only a range of 140° can be recorded. The missing tilt images later on result in a so-called missing-wedge in the 3D reconstruction (cryo-electron tomogram). In addition, the electron beam severely damages the sample during imaging, so only a low electron dose can be used to image a biological specimen. The low dose in combination with the presence of ice in the sample results in the acquired data being very noisy. Consequently, the identification of molecules within these reconstructions is a daunting task. Particle identification is however necessary as the particles need to be classified and averaged to determine high resolution structures. While cryoET has led to a large number of breakthroughs, providing hitherto unseen detail in the molecular architecture of cells,[2–4] the aforementioned challenges still hinder the widespread use of cryoET in the larger cell biology and structural biology community. In this context, the development of new reliable software tools is of paramount importance, which is however obstructed by the lack of sufficient accessible and annotated data to develop the software tools on.

SHREC simulator

To overcome the problem with the lack of data, in 2019, the annual *SHREC—3D Shape Retrieval Contest* included a new track titled *Classification in Cryo-Electron Tomograms*. The organizers of this track proposed a task of localization and classification of biological particles in cryo-electron tomograms. In the following years, experts from 3D object retrieval and 3D electron microscopy communities were invited to participate in the challenge. In order to ensure fair evaluation and comparable results across the submissions, the organizers created a dataset of ten physics-based cryo-electron tomogram simulations (9 train and 1 test tomogram, see Figure 1) for the contestants to train and evaluate their methods on. Each year, the results of the contesting methods were presented and compared.[5–7]

Unfortunately, simulating the tomographic data using SHREC is computationally very expensive. For a set of 10 small tilt-series

Table 1. Evaluation of simulated data quality by examining DeepFinder's performance on localization and classification tasks as a function of training data

Model	Training data	Data cost	Localiz. F1	Classification F1
DeepFinder	BENCHMARK	≈150 *h* (3 × CPU, 114 GB RAM)	0.815	0.581 (100 %)
DeepFinder	FAKET + fine-tuning	FAKET cost + manual labeling	0.821	0.565 (97 %)
DeepFinder	FAKET	≈12 *min* (1 × GPU, 40 GB VRAM)	0.800	0.533 (92 %)
		OR ≈80 *min* (8 × CPU, ≈6.5 GB RAM)		
DeepFinder	BASELINE	≈20 *s* (1 × CPU, 1 GB RAM)	0.813	0.441
TM-F			0.576	0.446
TM			0.372	0.470

The performance is evaluated on the same test tomogram. The results are shown in context with the performance of standard template matching algorithms (denoted TM and TM-F) reported in[7] for the same testing data. Performance is measured using the F1 score for localization and the F1 macro score for classification. Each of the DF results is an average of the test performances measured at the best epoch (based on validation) over 6 different random seeds. On the challenging classification task, DF trained using FAKET data simulated by our proposed method reaches 92% of the performance of DF trained using BENCHMARK data. It even reaches 97% when fine-tuned using a portion of BENCHMARK data. All this for a fraction of computational cost and without the need for a configuration protocol of the original simulation parameters used for crating BENCHMARK. The cost of data is given for the whole dataset comprising 10 tilt-series of shape (61 × 1024 × 1024), i.e., generating one such tilt-series with FAKET takes ≈70 seconds assuming a GPU is available. For comparison, a usual-sized tilt-series of shape (61 × 3500 × 3500), that is ≈12× larger in comparison to BENCHMARK, can be simulated under 10 min. For a comprehensive time and memory consumption profiling of FAKET, see Section Data S6.

(61 tilts of size 1024 × 1024) it took approx. 450 central processing unit (CPU) hours of computation on a node with 2× *Intel Xeon E5-2630 v4* CPUs. The implementation is able to utilize only 3 CPU cores per job in parallel and each job needs 114 GB of random-access memory (RAM). The memory consumption is also a reason why utilizing graphics processing units (GPU)s and simulating tilt-series of common sizes is not yet feasible (personal communication with the authors Gubins, I., and Chaillet, M.). This limits the scope of its applicability, mainly in data-hungry applications such as deep learning. Moreover, at the time of writing of this article, the source code of the simulator is not publicly available prohibiting more detailed comparisons, e.g., in terms of FLOPs.

Our contribution

In this paper, we propose FAKET, a fast and scalable data-driven method for simulating the forward operator of any cryo transmission electron microscope with the aim to generate synthetic micrographs or tilt-series. It was created, among other reasons, to generate fully labeled real-like data for training deep neural networks to solve tasks such as particle localization and (much more challenging) particle classification. Our method combines a noiseless simulated sample, see Figure 2, with additive noise, see Figure 3, and neural style transfer (NST) technique based on the study by Gatys L.A. et al.[8] to capture, using only unlabeled reference data, the structure of the complex noise introduced by TEM, see Figure 4. To carry out the NST, FAKET utilizes a pre-trained model, eliminating the need for users to train it themselves. That means users can use FAKET to simulate data from any TEM and under any configuration, assuming they posses unlabeled TEM data that could serve as a style reference. FAKET delivers data of quality nearly identical (in terms of their practical utility for subsequent tasks, rather than exact manifestation of the physical interactions between the sample and the transmission electron microscope) to the reference, at a cost only slightly higher than simply adding Gaussian noise to a noiseless synthetic sample (created using existing models of biological macromolecular structures, represented as Coulomb density volume). This makes it a practical, effective, and efficient choice for simulation.

For the purpose of evaluating our method, in the experiments presented within this article, we employed FAKET to mimic the behavior of the physics-based TEM simulator SHREC (due to the availability of ground truth). On top of evaluating the quality of our simulated data using standard image metrics (available in the supplement) that are of limited value in this context, we evaluated it directly on practically relevant downstream tasks by training DeepFinder (DF)[9]—a neural network specifically tailored to the task of particle localization and classification, see also DeepFinder.

In contrast to SHREC, our method accelerates the data generation process by a factor of 750 while using 33 times less memory (see Table 1). It also does not require any calibration protocol as other simulators, see Other simulators. The NST model does not need to be retrained to be used on new data, nor does it require labeled reference data. FAKET therefore has the potential to save experts countless hours of manual work in labeling their datasets. (Depending on the imaged sample, an expert may spend several hours of manual work per tilt-series per particle labeling the data. At the same time, certain smaller particles cannot be visually found at all, thus fully labeled tomograms do not really exist.) Moreover, FAKET is capable of simulating large tilt-series, which are common in experimental environments (about 12× larger than SHREC projections). For example, we generated a 61×3500×3500 tilt-series on a single *NVIDIA A100 40GB SXM4* GPU in less than 10 min. To achieve the reported speeds, a minimum of one GPU is recommended; CPU-only simulations are feasible and can be a valid option in specific situations, albeit slower. This advance makes it possible to train particle localization and classification networks from scratch or to pre-train networks that are later fine-tuned using manually labeled experimental data, see Section Data S3. On top of that, our method is open source and our experiments are reproducible.

Additionally, we include comparisons of the BENCHMARK and the proposed FAKET method with a simple BASELINE method based on

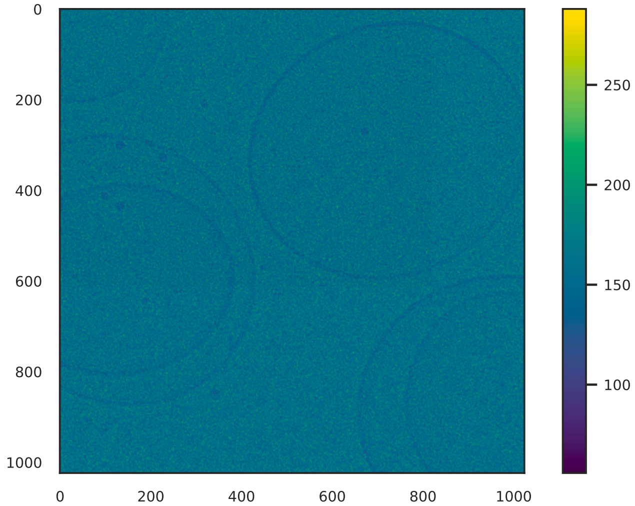


Figure 1. Simulated projection taken from SHREC 2021 dataset
Axes x and y correspond to width and height of the imaged grandmodel. Colorbar denotes simulated intensities in arbitrary units. See section Data S1 for side-by-side comparison with other projections.

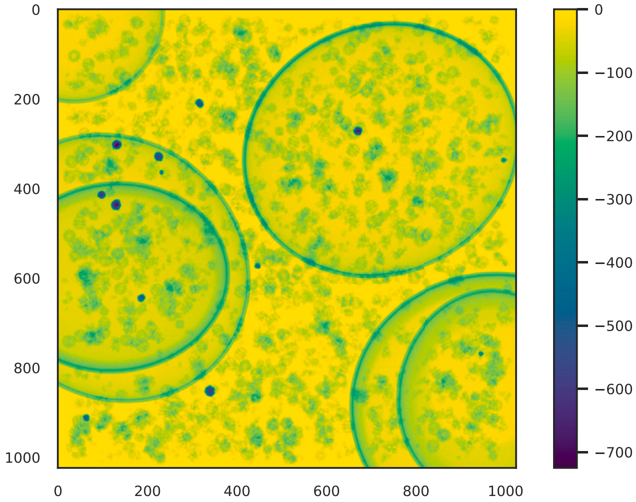


Figure 2. Noiseless projection used to create the input to our proposed method
Axes x and y correspond to width and height of the imaged grandmodel. Colorbar denotes intensities measured using Radon transform and negated such that particles have lower intensities than the background, as it is in the case of TEM which measures attenuation of electron beams. The particles are not embedded in any solvent (as if they were in vacuum instead of being embedded in ice), therefore the background appears much brighter than in the simulated projections. See section Data S1 for side-by-side comparison with other projections.

additive Gaussian noise, see Figure 3, aiming at providing comparative results in full practical range. The costly SHREC method defines an upper-boundary of performance, while the cheap addition of Gaussian noise establishes a lower boundary. This configuration enables us to position our method along this spectrum, showcasing FAKET's capability to offer results close to the upper-boundary while maintaining computational efficiency of a much simpler method. We hope our comparisons will be useful for practitioners who need to decide which method fits into their computational budget and for those seeking insightful understanding of the inherent trade-offs. The contribution of our method is further supported by an ablation study presented in Section Data S4, where we also offer insight into the potential limits of DF on the studied data by conducting experiments using completely noiseless simulated tomograms.

Related work

A similar idea in X-ray-based computed tomography angiography (CTA) was investigated in the study by Seemann M. et al.[10] The authors focused on solving a lumen segmentation task. However, due to the very different nature of the samples imaged in CTA (the objects of interest in CTA are on average two orders of magnitude larger in relation to the size of the tomogram than those in cryoET, where the particles often span only tens of voxels), it was not clear whether a similar NST based framework could be successfully applied in cryoET to image nanoscale particles. In addition, the article does not provide source code and does not document the experiments in enough detail for us to be able to reproduce the results or adapt the method to the cryoET domain.

Other simulators

The value of simulated data in cryoEM is well recognized and micrograph simulation has been attempted several times. In first approximation, a single cryoEM micrograph is the projection of a 3D object, convolved with the electron microscope's point spread function. Additionally, the overall process comprises several sources of noise coming from the nature of the sample, the microscope, and the imaging procedure which are hard to model accurately. For a better overview of the attempts, it is necessary to mention TEM simulators that were developed in the past decades to simulate micrographs in cryo conditions.

Earlier works provided fast simplistic models mostly based on additive white Gaussian noise (this motivated our choice of baseline) or colored noise.[11] Other works originated from the insights into the physics of TEM image formation and advanced the simulations by modeling various sources of noise, e.g., an improvement in modeling the structural noise was proposed in *TEM Simulator* (C).[12] Another improvement was done in *InSilicoTEM* (MATLAB) presented in the study by Vulovic M. et al.[13] by implementing the multislice method originally proposed in the study by Cowley J.M. et al.[14] This makes it the next most relevant simulator related to our work after SHREC which is, as many others, also based on the same multislice method (this motivated our choice of benchmark). To the best of our knowledge, the most recent improvement presented in the study by Himes B. and Grigorieff N.[15] is distributed as a part of *cisTEM* package (C++). It introduced frozen plasmon method to explicitly model spatially variable inelastic scattering processes in cryo-electron microscopy. The aforementioned works, based on the same multislice method, however, suffer from a heavy computational burden or are limited to simulations of a single molecular complex. Despite these limitations, they represent a set of precise advanced physics-based simulators and as such also provide a comprehensive literature survey referring the reader to a body of detailed

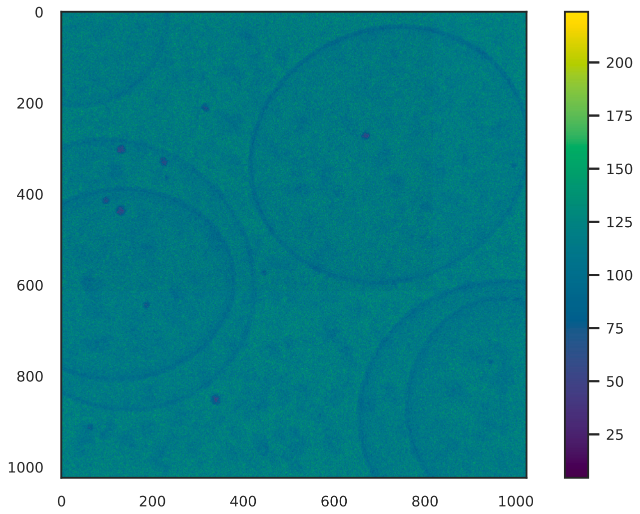


Figure 3. BASELINE projection created by adding Gaussian noise to the noiseless projection

Axes x and y correspond to width and height of the imaged grandmodel. Colorbar denotes simulated intensities in arbitrary units. Please note that this projection is not exactly the same as in Figure 1 or in Figure 4, see also the explanation in the caption of Figure 4 and section Data S1 for a side-by-side comparison with other projections.

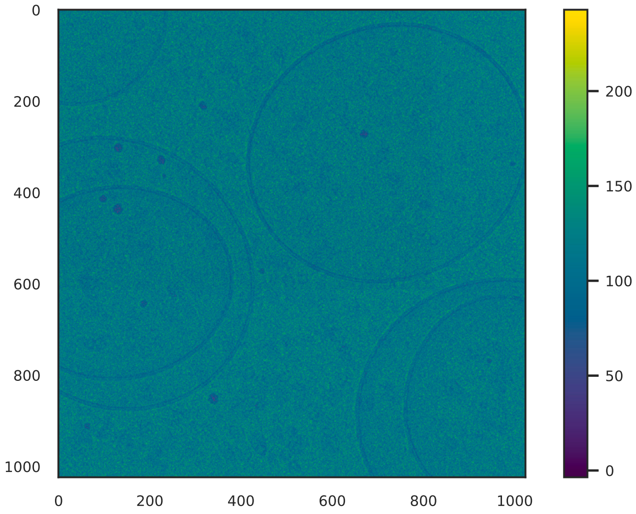


Figure 4. FAKET projection output by our method

Axes x and y correspond to the grandmodel's width and height respectively, and colorbar indicates simulated intensities in arbitrary units. Though visually similar to Figure 3, there is a subtle difference that has a significant impact on DF's performance. This similarity complicates comparison of projections from various simulators using currently available image metrics. See section Data S1 for a side-by-side comparison with other projections.

resources about modeling the image formation in transmission electron microscopy.

In material sciences, GPU accelerated simulators such as *MULTEM* (C++, CUDA),[16,17] *abTEM* (Python),[18] or *Prismatic* (C++, CUDA)[19] have emerged. However, as reviewed in in the study by Kirkland E.J.,[20] advanced TEM simulators used in material sciences require atomic models of the background and the entire specimen. Moreover, in cryoEM, the sample and its interaction with the electron beam is fundamentally different. While in material science samples are often only a few layers of atoms thick and the imaged atoms have strong interaction with the electron beam, cryoEM samples are thousands of atom layers thick, are less ordered, and only weakly interact with the electron beam. Additionally, cryoEM samples can only withstand little radiation before complete destruction resulting in much lower signal to noise ratios. Due to these differences, such simulators are of limited use for macromolecular biological specimen simulations that we experiment with in this article. Finally, in contrary to our approach, all aforementioned methods require precise calibration protocols for setting the values of simulation parameters.

Structure of this article

To build our proposed method, presented in method details (referred to as FAKET), we used a data-driven approach (i.e., no calibration protocol is needed). More information about the data we used is in Data. With our method, we managed to successfully approximate the input-output behavior, on par with the SHREC simulator (referred to as BENCHMARK) but for a fraction of its computational cost. To experimentally prove the need for our method, we also compared it to a simple addition of Gaussian noise (referred to as BASELINE) that was naturally very fast to compute, but did not lead to comparable results on the downstream tasks.

To evaluate the methods, we used the simulated data by each of them as a training set for the DeepFinder neural network (more in DeepFinder). The network was trained to solve two evaluation tasks proposed in SHREC challenge. The final comparison was done by observing the models' performances on the test tomogram chosen by SHREC. Detailed description of the evaluation is in Evaluation. More on experiments and results is in Results section. Procedures of each method are compared in Figure 5. This paper is accompanied with source code, the results are fully-reproducible, and the full experiment results are available in the repository – https://github.com/paloha/faket/.

Side-by-side comparison of all projections used in this article is available in section Data S1. The process of re-creating the BENCHMARK data is detailed in section Data S2. Fine-tuning experiments that further improved the DF performance are described in section Data S3. Additional experiments, NST ablation study, and investigation of DF limits are presented in section Data S4. Further details on the performance of studied models on the particle classification task are provided in section Data S5.

Key findings

(1) FAKET successfully utilizes NST to simulate cryo-electron micrographs or tilt-series of common sizes. (2) FAKET produces realistic TEM simulations of quality nearly identical to the unlabeled reference data. (3) Cost of FAKET simulation is only slightly higher than simply adding Gaussian noise while significantly lower than that of complex physics-based simulators. (4) Apart from unlabeled reference data, FAKET does not require further training or calibration. (5) FAKET is a practical and efficient tool for simulating fully labeled training data for deep learning.

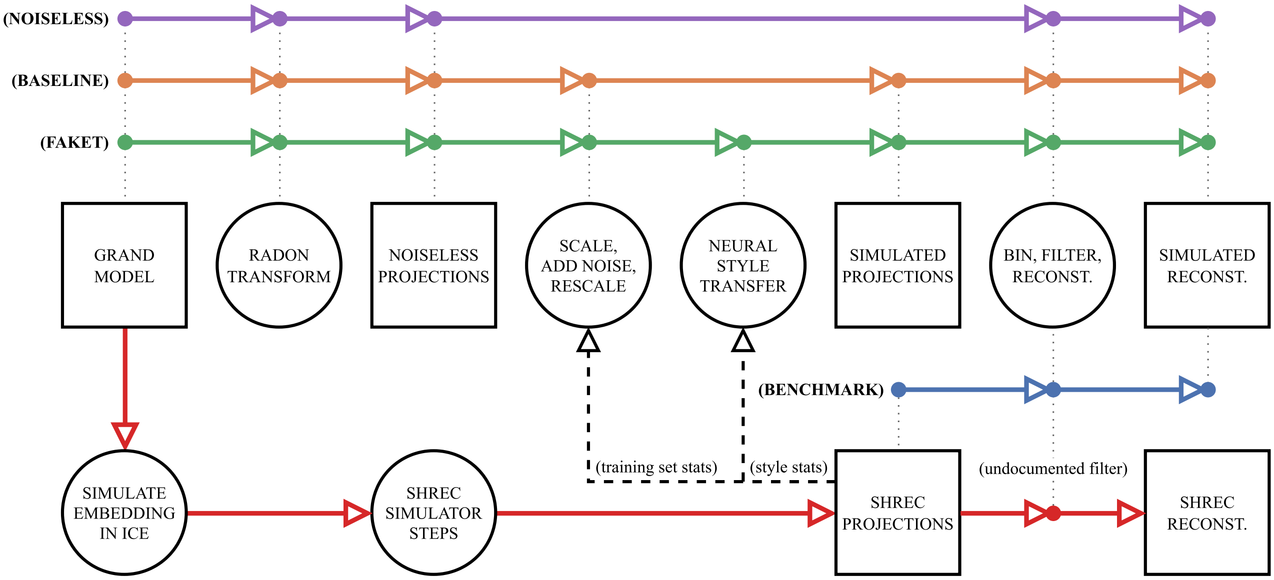


Figure 5. Diagram of steps to simulate the BENCHMARK, BASELINE, FAKET, and noiseless projections and reconstructions

Red arrows highlight steps to reproduce SHREC data from which we use the last tomogram for testing. All methods except SHREC were filtered using a reverse-engineered filter (see section Data S2) because the SHREC filtering step is under-documented. The style projections never feature the same contents as the simulated ones (see section Data S1). Grandmodels, noiseless artificial samples containing randomly scattered particles, were created using existing models of biological macromolecular structures, represented as Coulomb density volumes.

RESULTS

Comparison of DeepFinder's performance on localization and classification tasks as a function of training data is presented in Table 1. The mean performances per epoch with 68% and 95% confidence intervals (CIs) are presented in Figures 6 and 7. Results of additional experiments can be found in section Data S3, Data S4, and Data S5.

The performance of DF trained on BENCHMARK data turned out to be the best in both localization and classification tasks, as expected, since the testing tomogram was reconstructed from the exactly same tilt-series as BENCHMARK training set. In practice, however, it is infeasible to have such favorable conditions as fully labeled tomograms are not existent. Most biological studies focus on one or a subset of molecules to be labeled. And since the labeling process is laborious and requires domain experts, it is also expensive. With that in mind, we can hardly expect to have a training set of tomograms from exactly the same distribution as the tomograms we are interested in. Nevertheless, this result serves us, as the name suggests, as a benchmark that we are trying to reach with our proposed methods.

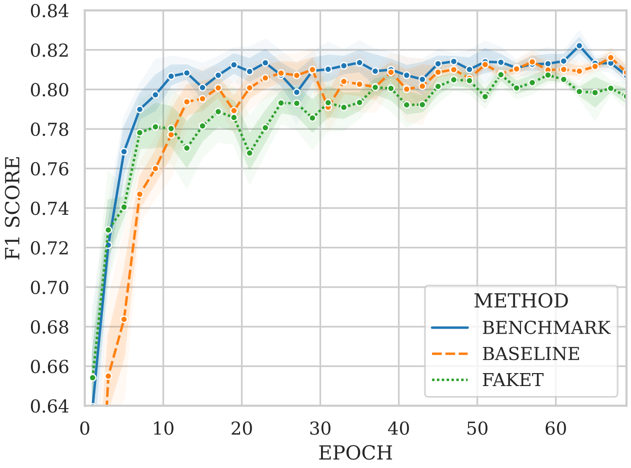


Figure 6. Performance on localization task

Darker and lighter regions show 68% and 95% CI.

The highest BENCHMARK F1 score of 81.5% (68% CI [81.1%, 82.0%]) on localization task and 58.1% (68% CI [57.9%, 58.3%]) on classification task on average across 6 different random seeds was achieved after 65 ± 4 epochs of training.

To put these numbers in perspective, we investigated the limits of DF performance by training and evaluating on completely noiseless data, see section Data S4 for more details. In this setting, DF achieved F1 score of 83.2% (68% CI [82.4%, 84.0%]) on localization task and 72.5% (68% CI [70.5%, 74.6%]) on classification task.

Our fastest method of simulating the projections and subsequently the tomograms, BASELINE, scored 81.3% (68% CI [81.1%, 81.6%]) on localization task. It was a surprise that such a simple method led to so high localization performance, considering that it requires ≈150 *h* to create the BENCHMARK projections, while it takes almost no time to create the BASELINE projections. This method can be therefore regarded as the "poor-man's" choice in settings where the computational budget is very limited, or in settings where this task has to be performed repeatedly many times. Unfortunately, the classification

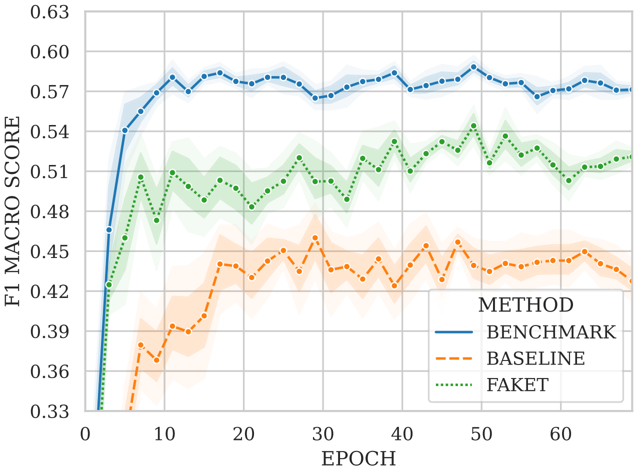


Figure 7. Performance on classification task
Darker and lighter regions show 68% and 95% CI.

performance of 44.1% (68% CI [43.2%, 45.1%]) was rather poor, not surprisingly, as classification is an inherently harder task, cf.[7]

Our proposed method FAKET, based on additive noise (using our advanced estimation of in-projections noise) and subsequently neural style transfer, with on average 80.0% (68% CI [79.8%, 80.2%]) F1 score on localization task performed on par with the BENCHMARK and significantly outperformed standard template matching algorithms. The best performance on the classification task was achieved after 65 ± 4 epochs with a score of 53.3% (68% CI [52.8%, 53.8%]). With this result, FAKET matched the BENCHMARK to 92% while reducing the cost of data generation by a factor of 750 and using 33 times less memory. Multi-class classification performance certainly can not be reduced only to one number and it is important to consider also other metrics, such as per-class classification report and confusion matrices. The per-class classification performance is shown in Figure 8 and we provide the confusion matrices in section Data S5. The full performance reports are available in the accompanying repository for enthusiastic readers.

In section Data S3, we present further experimental results showcasing the fine-tuning of FAKET models using a subset of BENCHMARK data. This approach further narrows the performance gap between the FAKET and BENCHMARK models, achieving 97% of the BENCHMARK model's classification performance, while exceeding its performance on localization task. These results might be valuable for cryoET practitioners who are seeking to maximize the performance of their models and have, or can get a small but representative sample of their data labeled.

DISCUSSION

In this paper, we proposed FAKET, a fast and scalable data-driven method for simulating the forward operator of any electron microscope based on additive noise and neural style transfer. The proposed method can be used for generating synthetic cryo-electron micrographs or tilt-series that closely approximate the quality of reference TEM data, at a computational cost only marginally higher than that of simply adding Gaussian noise. The data can be used to train deep neural networks to solve tasks such as particle localization and (much more challenging) particle classification. The field of cryo-electron tomography currently suffers from the lack of sufficient amounts of annotated data, and the proposed method aims to solve this problem.

In this study, we evaluated FAKET as a surrogate model that mimics the behavior of the physics-based TEM simulator SHREC, while being drastically less computationally expensive, both in terms of time and memory. It accelerates the data generation process by a factor of 750 while using 33 times less memory, making the generation of thousands of tilt-series feasible. Moreover, it is capable of simulating large tilt-series, which are common in experimental environments. For example, we generated a 61×3500×3500 tilt-series on a single *NVIDIA A100 40GB SXM4* GPU in less than 10 min. It also does not require any calibration protocol, it does not need to be retrained to be used on new data, nor does it require labeled reference data. The method is open source, and the experiments are reproducible.

The quality of our approximations was evaluated using the DeepFinder network, which emerged from the SHREC challenge as one of the most successful. The results showed that the

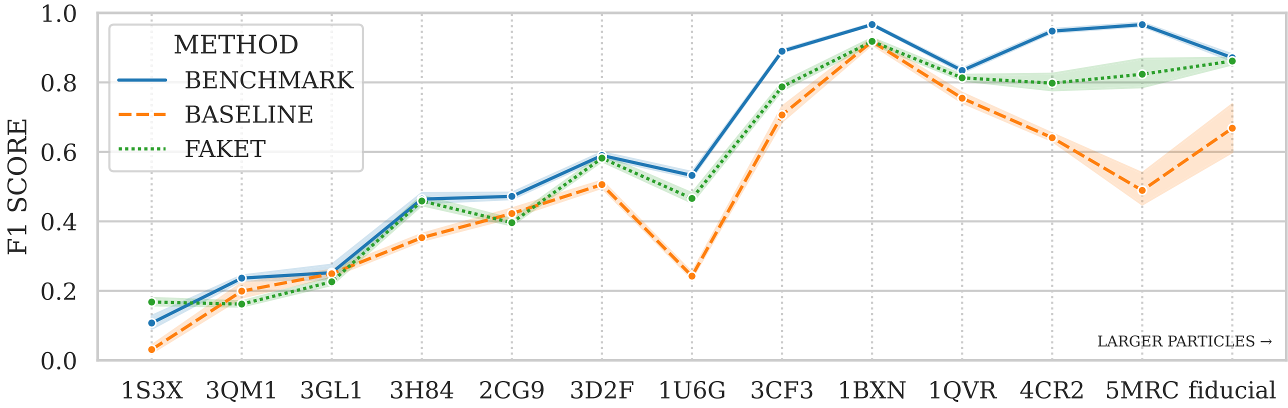


Figure 8. Per-class performance on classification task
Darker and lighter regions show 68% and 95% CI.

performance of models trained using our approximations is on par with the BENCHMARK method on localization task and reached 92% of its performance on classification task while significantly outperforming standard template matching algorithms. When further fine-tuned using a portion of BENCHMARK data, the classification performance was improved to 97%.

This advancement simplifies the generation of fully labeled, high-quality synthetic tilt-series that resemble experimental TEM data requiring analysis. This simulated data can either be used to train particle localization and classification neural networks from scratch, or serve as pre-training data for networks that will be fine-tuned with manually labeled experimental data later. Researchers investigating reconstruction algorithms can also benefit from our simulator since the availability of ground truth allows for effortless evaluation of their novel methods. However, when simulating data for use cases other than training neural networks for particle localization or classification, it is important to take into account that no explicit knowledge of the underlying physics phenomena of TEM is built into the simulator.

In future work, we will focus on further validation of our method using experimental TEM data. Additionally, we plan to improve the method by replacing the VGG-19 network that was pretrained on natural images, with a state-of-the-art vision network pretrained on cryoEM data. Moreover, we aim to provide the community of practitioners with a fully functional and easy-to-use piece of software for generating their synthetic datasets. Either based on chosen particles from Protein Data Bank, or using already available whole-cell models. The goal is to enable more accurate and efficient data analysis while also making the process more accessible to researchers in the field. We hope this advancement will serve as the basis for development of new computational methods in cryoEM and cryoET.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources should be directed to and will be fulfilled by Dr. Pavol Harar (pavol.harar@ista.ac.at, or find the current contact using ORCID: 0000-0001-5206-1794).

Materials availability

This study did not generate new unique materials.

Data and code availability

All data, the original source code of FakET, and additional information needed to reproduce this study are publicly available as of the date of publication. DOIs and URLs are listed in the key resources table. Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

ACKNOWLEDGMENTS

The IMP and D.H. are generously funded by Boehringer Ingelheim. We thank Julius Berner from the Mathematical Data Science group @ UniVie, Ilja Gubins and Marten Chaillet from the SHREC team, and the members of the Haselbach lab for helpful discussions.

AUTHOR CONTRIBUTIONS

Conceptualization, P.H. and D.H.; methodology, software, validation, formal analysis, investigation, P.H. and L.H.; data curation and visualization, P.H.; resources, P.G.; writing - original draft, P.H., P.G., and D.H.; writing - review and editing, P.H. and D.H.; supervision, P.G. and D.H.; funding acquisition, P.G. and D.H.

DECLARATION OF INTERESTS

The authors declare no competing interests.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- METHOD DETAILS
  - ○ Additive noise (BASELINE)
  - ○ Neural style transfer (FAKET)
- QUANTIFICATION AND STATISTICAL ANALYSIS
  - ○ Data
  - ○ Evaluation
  - ○ DeepFinder

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.str.2025.01.020.

Received: July 18, 2023
Revised: November 2, 2023
Accepted: January 17, 2025
Published: February 12, 2025

REFERENCES

1. Turk, M., and Baumeister, W. (2020). The promise and the challenges of cryo-electron tomography. FEBS Lett. *594*, 3243–3261. https://doi.org/10.1002/1873-3468.13948.
2. Zimmerli, C.E., Allegretti, M., Rantos, V., Goetz, S.K., Obarska-Kosinska, A., Zagoriy, I., Halavatyi, A., Hummer, G., Mahamid, J., Kosinski, J., and Beck, M. (2021). Nuclear pores dilate and constrict in cellulo. Science *374*, eabd9776. https://doi.org/10.1126/science.abd9776.
3. O'Reilly, F.J., Xue, L., Graziadei, A., Sinn, L., Lenz, S., Tegunov, D., Blötz, C., Singh, N., Hagen, W.J.H., Cramer, P., et al. (2020). In-cell architecture of an actively transcribing-translating expressome. Science *369*, 554–557. https://doi.org/10.1126/science.abb3758.
4. Mahamid, J., Pfeffer, S., Schaffer, M., Villa, E., Danev, R., Cuellar, L.K., Förster, F., Hyman, A.A., Plitzko, J.M., and Baumeister, W. (2016). Visualizing the molecular sociology at the HeLa cell nuclear periphery. Science *351*, 969–972. https://doi.org/10.1126/science.aad8857.
5. Gubins, I., Schot, G.v. d., Veltkamp, R.C., Förster, F., Du, X., Zeng, X., Zhu, Z., Chang, L., Xu, M., Moebel, E., et al. (2019). Classification in Cryo-Electron Tomograms. In Eurographics workshop on 3d object retrieval, S. Biasotti, G. Lavoué, and R. Veltkamp, eds. (The Eurographics Association). https://doi.org/10.2312/3dor.20191061.
6. Gubins, I., Chaillet, M.L., van der Schot, G., Veltkamp, R.C., Förster, F., Hao, Y., Wan, X., Cui, X., Zhang, F., Moebel, E., et al. (2020). Shrec 2020: Classification in cryo-electron tomograms. Comput. Graph. *91*, 279–289. https://doi.org/10.1016/j.cag.2020.07.010.
7. Gubins, I., Chaillet, M.L., Schot, G.v. d., Trueba, M.C., Veltkamp, R.C., Förster, F., Wang, X., Kihara, D., Moebel, E., Nguyen, N.P., et al. (2021). SHREC 2021: Classification in Cryo-electron Tomograms. In Eurographics workshop on 3d object retrieval, S. Biasotti, R.M. Dyke, Y. Lai, P.L. Rosin, and R.C. Veltkamp, eds. (The Eurographics Association). https://doi.org/10.2312/3dor.20211307.
8. Gatys, L.A., Ecker, A.S., and Bethge, M. (2016). Image style transfer using convolutional neural networks. In Proceedings of the IEEE Conference on

Computer Vision and Pattern Recognition (CVPR) (IEEE / CVF), pp. 2414–2423. https://doi.org/10.1109/CVPR.2016.265.

9. Moebel, E., Martinez-Sanchez, A., Lamm, L., Righetto, R.D., Wietrzynski, W., Albert, S., Larivière, D., Fourmentin, E., Pfeffer, S., Ortiz, J., et al. (2021). Deep learning improves macromolecule identification in 3d cellular cryo-electron tomograms. Nat. Methods *18*, 1386–1394. https://doi.org/10.1038/s41592-021-01275-4.
10. Seemann, M., Bargsten, L., and Schlaefer, A. (2020). Data augmentation for computed tomography angiography via synthetic image generation and neural domain adaptation. Curr. Dir. Biomed. Eng. *6*, 20200015. https://doi.org/10.1515/cdbme-2020-0015.
11. Scheible, P., Sazzed, S., He, J., and Wriggers, W. (2021). Tomosim: Simulation of filamentous cryoelectron tomograms. Proceedings *2021*, 2560–2565. https://doi.org/10.1109/BIBM52615.2021.9669370.
12. Rullgård, H., Öfverstedt, L.-G., Masich, S., Daneholt, B., and Öktem, O. (2011). Simulation of transmission electron microscope images of biological specimens. J. Microsc. *243*, 234–256. https://doi.org/10.1111/j.1365-2818.2011.03497.x.
13. Vulović, M., Ravelli, R.B.G., van Vliet, L.J., Koster, A.J., Lazić, I., Lücken, U., Rullgård, H., Öktem, O., and Rieger, B. (2013). Image formation modeling in cryo-electron microscopy. J. Struct. Biol. *183*, 19–32. https://doi.org/10.1016/j.jsb.2013.05.008.
14. Cowley, J.M., and Moodie, A.F. (1957). The scattering of electrons by atoms and crystals. i. a new theoretical approach. Acta Crystallogr. *10*, 609–619. https://doi.org/10.1107/S0365110X57002194.
15. Himes, B., and Grigorieff, N. (2021). Cryo-tem simulations of amorphous radiation-sensitive samples using multislice wave propagation. IUCrJ *8*, 943–953. https://doi.org/10.1107/S2052252521008538.
16. Lobato, I., and Van Dyck, D. (2015). 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. https://doi.org/10.1016/j.ultramic.2015.04.016.
17. Lobato, I., Van Aert, S., and Verbeeck, J. (2016). Progress and new advances in simulating electron microscopy datasets using multem. Ultramicroscopy *168*, 17–27. https://doi.org/10.1016/j.ultramic.2016.06.003.
18. Madsen, J., and Susi, T. (2020). Abtem: Ab initio transmission electron microscopy image simulation. Microsc. Microanal. *26*, 448–450. https://doi.org/10.1017/S1431927620014701.
19. Pryor, A., Ophus, C., and Miao, J. (2017). A streaming multi-gpu implementation of image simulation algorithms for scanning transmission electron microscopy. Adv. Struct. Chem. Imaging *3*, 15. https://doi.org/10.1186/s40679-017-0048-z.
20. Kirkland, E.J. (2020). Advanced Computing in Electron Microscopy (Springer Nature). https://doi.org/10.1007/978-3-030-33260-0.
21. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, highperformance deep learning library. Adv. Neural Inf. Process. Syst. *32*, 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.
22. Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. Preprint at arXiv. https://doi.org/10.48550/arXiv.1409.1556.
23. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., and Bourne, P.E. (2000). The Protein Data Bank. Nucleic Acids Res. *28*, 235–242. https://doi.org/10.1093/nar/28.1.235.
24. Hrabe, T., Chen, Y., Pfeffer, S., Cuellar, L.K., Mangold, A.-V., and Förster, F. (2012). Pytom: A python-based toolbox for localization of macromolecules in cryo-electron tomograms and subtomogram analysis. J. Struct. Biol. *178*, 177–188. https://doi.org/10.1016/j.jsb.2011.12.003.
25. Van derWalt, S., Schönberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Gouillart, E., and Yu, T.; scikit-image contributors (2014). Scikit-image: Image processing in python. PeerJ *2*, e453. https://doi.org/10.7717/peerj.453.
26. Müller, P. (2013). Radontea: Python algorithms for the inversion of the radon transform. https://pypi.python.org/pypi/radontea/.
27. Mastronarde, D.N., and Held, S.R. (2017). Automated tilt series alignment and tomographic reconstruction in imod. J. Struct. Biol. *197*, 102–113. https://doi.org/10.1016/j.jsb.2016.07.011.
28. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Advances in neural information processing systems, I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds. (Curran Associates, Inc) https://proceedings.neurips.cc/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf.
29. Chollet F. (2015). Keras. https://keras.io/getting_started/faq/#how-should-i-cite-keras@misc{chollet2015keras.
30. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems [Software available from tensorflow.org]. https://www.tensorflow.org/.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
2021 SHREC challenge dataset described by Gubins et al., 2021 in https://doi.org/10.2312/3dor.20211307. ● shrec2021_original_groundtruth.zip ● shrec2021_full_dataset.zip	Gubins et al.	https://doi.org/10.34894/XRTJMA (version 2.0)
Software and algorithms		
Accompanying repository of FakET: Simulating Cryo-Electron Tomograms with Neural Style Transfer by Harar et al., 2023. The repository contains all source code needed to run the method or reproduce all experiments exactly as presented in the paper.	This paper	https://github.com/paloha/faket/releases/tag/2304.02011v3
2021 SHREC challenge eval.py evaluation script used by Gubins et al., 2021 in https://doi.org/10.2312/3dor.20211307. This script was adjusted for this study and is available from the accompanying repository of FakET linked in this table.	Gubins et al.	https://doi.org/10.34894/XRTJMA (version 2.0)
Implementation of the DeepFinder v1.0 method published by Moebel et al. 2021 in https://doi.org/10.1038/s41592-021-01275-4. This method was adjusted for this study and is available from the accompanying repository of FakET linked in this table.	Moebl et al.	https://gitlab.inria.fr/serpico/deep-finder/-/tree/fa6a0c2b7b792e888d0619c847302f0954caead2/deepfinder
Implementation of the Neural Style Transfer method described by Gatys et al. 2016 in https://doi.org/10.1109/CVPR.2016.265. This method was adjusted for this study and is available from the accompanying repository of FakET linked in this table.	Katherine Crowson	https://github.com/crowsonkb/style-transfer-pytorch/commit/1107fe68639a59bd54bcda018e25dd770819ab19

METHOD DETAILS

The methods described in this section represent surrogate models that mimic the behaviour of the aforementioned SHREC physics-based simulator. The methods are to be applied in the projection space, i.e. all of them take noiseless (and ice-less) projections as inputs, shown in Figure 2, and produce real-like looking projections by matching their appearance with a target “style” projections. The outputs are ready to be reconstructed into final tomogram volumes using a reconstruction algorithm of choice. There are multiple reasons why we decided, and why it is reasonable, to model the simulator’s behaviour in projection space. Firstly, different reconstruction algorithms produce different types of artifacts and are suitable in different situations. Therefore, simulating only projections means the models do not have to mimic the artifacts crated by the reconstruction algorithm (e.g. smearing due to the missing wedge). Moreover, this approach does not limit the practitioners in their choice of the reconstruction algorithm, and opens the possibility for researchers to also use the methods in research of novel reconstruction algorithms. But most importantly, simulating in projection space means the models need to process and produce an order-of-magnitude less data points, as compared to simulating the final reconstruction volumes. That means, simulating $\theta \times N^2$ data points instead of N^3. In our case, $\theta = 61$ and $N = 1024$. In practice, θ stays approximately the same, but N is usually 3 – 4 × larger.

Additive noise (BASELINE)

Before moving on to the development of more complicated methods, we wanted to see how a very simple method involving tilt-dependent scaling and the addition of Gaussian noise would approximate the target projections. The modality of projections produced by this method is referred to as BASELINE projections in the whole text. First, we shifted and scaled each tilt of the noiseless input projections separately according to the average mean and standard deviation of each tilt within the training set. Through this tilt-dependent scaling, we attempt to model different degrees of attenuation as a function of tilt angle - the more extreme the tilt angle, the greater the attenuation due to the longer electron beam trajectory.

Next, we simply add Gaussian noise, as this is the natural choice of practitioners when they need to quickly create some simulated tilt-series. Of course, this raises the question of how much noise to add. In reality, practitioners would probably visually choose a value for σ that gives a similar signal-to-noise ratio. However, we opted for a more objective procedure by subtracting the noiseless content from the target projections to extract the noise and measure its statistics. With SHREC data, where precise ground truth is available, this task is less challenging than with real data, where we lack ground truth for the entire tomogram. So in a real scenario we would have to select regions for which we can estimate the ground truth and calculate the noise statistics only for these regions instead of using the whole tilt values. After adding the Gaussian noise, the resulting volume was scaled to match the average mean and standard deviation of the training set with respect to tilt. The steps for creating projections and consequently reconstructions using this method are shown in Figure 5, and an example of a simulated projection can be seen in Figure 3. The calculation of this modality takes only a few seconds using a single CPU.

Neural style transfer (FAKET)

To capture the noise structure of the BENCHMARK projections more closely, in FAKET we devised a more elaborate method of estimating the noise statistics as opposed to the simpler one used in BASELINE. In this case, we estimate the noise statistics for each tilt in the whole training set separately and fit a second-degree polynomial over the averages. This process captures the average noise statistics as a function of tilt angle based on the information from the whole training set. After adding the better estimated Gaussian noise, we obtained projections which we will refer to as *noisy projections*. To further adapt those projections, we used the Neural Style Transfer technique implemented in PyTorch framework[21] and introduced in.[8]

NST was built to render the semantic content of natural images in different styles. At its core lies VGG net,[22] a convolutional neural network optimized for object recognition and -localization. Within the NST framework, the VGG-19 model is used for extracting the content and style representations of so-called ''content'' and ''style'' images, which are provided as inputs. NST then iteratively updates the output image to simultaneously match the content representation and the style representation of the provided inputs at multiple scales. The NST technique is described in the aforementioned paper in great detail and we encourage the reader to consult it if any questions would arise. The VGG-19 model used in FAKET was pretrained on ImageNet data set of natural images, eliminating the need for users to train the model themselves. For those interested in further fine-tuning the NST model (benefits of which are still under investigation by the authors), guidelines are available on the PyTorch website. Fine-tuning would necessitate partially labeled data, unless an autoencoder approach is adopted, which would require no labels.

Cryo-electron projections are not natural images as those used to pre-train the VGG net. It would be therefore surprising if the NST provided us with desired results ''out-of-the-box''. The first experiments with NST using the noiseless projections as content images and BENCHMARK projections as style images were disappointing due to numerous strong artifacts scattered apparently randomly over the adapted projections. Search over the space of hyper-parameters did not result in satisfactory output even after 10 thousand iterations. Further experiments with BASELINE projections as content images performed poorly on the evaluation task, especially on localization of smaller particles. It might be useful for the readers to know that before we even decided to work in projection space, our very first experiments were done in reconstruction space, but this idea had to be quickly abandoned due to the poor performance, failure to transfer the artifacts of reconstruction algorithm and computational infeasibility.

In order to obtain the desired results, we adjusted the NST to our specific needs. Firstly, we adjusted the code to accept sequences of 1-channel floating point arrays as inputs to improve the speed of processing. Next, we rid the code of conversions associated with handling RGB images in order to preserve the floating point precision of our data. And finally, we implemented the support for other than random initialization or initialization with the content which was a crucial change to produce the desired results. We initialized the NST with *noisy projections*, used *noisy projections* with 25% of noise as content, and provided the associated BENCHMARK projections as style images. We would like to point out that the associated style images were taken from a different training tomogram, therefore do not feature the same content. I.e. it is not possible to simply minimize the element-wise mean absolute error to get the desired output.

From the NST pipeline, we only used the 1024x1024 scale because downsampling to smaller scales combined with anti-aliasing used in NST to transition between the scales is well suited for natural images and lots of NST iterations but not for our use case on scientific data. Using new initialization and slightly noisy content images allowed us to increase the learning rate of the NST optimizer, so we were able to produce visually persuasive outputs in just one iteration, as seen in Figure 4.

This method is more involved than just adding the noise, but produces better results, is very fast due to our changes of NST initialization and hyper-parameters, requires less domain-specific knowledge than implementing the forward operator, and can be used in real-world scenarios. Computing this modality of data, including the steps to create the *noisy projections* and *content projections* for 10 tomograms took only $\approx$ 12 minutes on a single *NVIDIA A100 40GB SXM4* GPU, which on average represents a 750× speedup in comparison to BENCHMARK. This method also needed only $\approx$ 3.5 GB of GPU memory per tomogram as opposed to 117GB of RAM which was necessary for BENCHMARK. The GPU memory usage of our method could be even further optimized. This represents 33 times less memory usage. Certainly, it is not possible to do a head-to-head comparisons between GPU memory and RAM, but this information can be used to estimate the hardware requirements. Moreover, as opposed to the BENCHMARK method, it is possible to scale our method to volumes that match usual sizes of experimental data on still reasonable hardware.

QUANTIFICATION AND STATISTICAL ANALYSIS

Data

For the purposes of this paper, we used the latest version of the SHREC data set from the year 2021. More specifically, we based the presented methods on full-resolution grandmodels (synthetic volumes containing randomly scattered particles) and simulated projections from the *SHREC 2021 additional* data set that was made publicly available (more info at www.shrec.net/cryo-et, download version 2.0 from https://doi.org/10.34894/XRTJMA). An example visualization of a simulated projection using the SHREC simulator is depicted in Figure 1. We use the unchanged simulated projections as a benchmark which we try to approximate with our proposed method.

The steps to create the simulations were described in.[7] However, at the time of writing of this paper, the implementation is not publicly available and from the description of the steps it is clear that the method is, and rightfully so, very elaborate, and not at all elementary to reimplement (also due to various steps being under-documented, see Section Data S2).

On the following lines, we briefly summarize the steps needed to create the SHREC data set. The authors first constructed 3D ground-truth specimens (grandmodels) along with annotations. Each specimen contained uniformly distributed and rotated protein instances (represented as Coulomb density) of varying size and structure from the Protein Data Bank (PDB),[23] as well as membranes and gold fiducials, which are commonly found in tomograms. Then, they simulated a layer of amorphous ice into the grandmodels before rotating them over 61 evenly-spaced tilt angles, ranging from -60° to $+60^\circ$, in order to be projected. The noiseless projections were produced using their own implementation of a TEM simulator based on the multislice approach presented in.[13] Next, they sampled from a Poisson distribution with a specific electron dose to obtain the final electron counts (files labeled as *projections*). After that, they randomly shifted the projections to model the tilt misalignment and scaled the amplitudes in Fourier space using information about amplitudes from experimental images in order to increase their similarity (files labeled as *projections_unbinned*). In the end, they obtained the final images (files labeled as *reconstruction*) by reconstructing the 2× binned projections using weighted-backprojection algorithm from a private version of the PyTom package.[24]

We chose the SHREC data set as it is established, well-executed, downloadable, and allows researchers to compare their results with previous works without the need to reproduce all results from scratch. In this paper, we are not focusing on solving the proposed tasks, instead, we are using them as evaluation metrics to measure the performance of our proposed method. However, the availability of the data and descriptions of the methods made this research feasible.

As stated earlier, we used the full-resolution grandmodels from the SHREC data set to create our own noiseless projections using Radon transform, as implemented in the *scikit-image* library for image processing in Python.[25] We computed our own noiseless projections for three primary reasons. Firstly, SHREC only supplied full-resolution noiseless projections embedded within the simulated ice layer, a feature we also aimed to incorporate in our surrogate. Secondly, we intended to provide access to all implementation steps within the code-base. Lastly, it was necessary to produce noiseless reconstructions that could later be used to estimate the performance boundaries of DF on this data set. We also had to omit the usage of the reconstructions provided by SHREC because the exact configuration of all the steps is not published and the version of the PyTom software package used to create this data is not public. Also, the public version is not yet properly documented or straightforward to use. Thus, the only feasible option to ensure comparable results across our experiments was to create our own reconstructions from all the newly created projections described in Method details. Nevertheless, we put a lot of effort into matching our reconstructions with the ones originally published in SHREC. We created them using the RadonTea Python package[26] using a custom filter described in Section Data S2. For additional flexibility, we also implemented support for tomographic reconstructions using the well-known IMOD package.[27] To ensure comparability also with previously published results, we used the original SHREC *model_9* reconstruction as a test tomogram for all presented methods as it was done in all the past challenges.

Evaluation

To evaluate the quality of the proposed simulated projections and subsequently tomograms, no widely accepted metrics exist,[10] and it is not clear how well the standard metrics such as the mean squared error, or more elaborate metrics such as the Fréchet inception distance,[28] relate to the performance on tasks of interest to the practitioners.

We therefore trained a randomly initialized DF neural network to solve two evaluation tasks proposed in the SHREC 2021 challenge, namely particle localization and classification. These tasks are of utmost importance to practitioners in determining the structure of proteins and macromolecular complexes. We argue that evaluation on a task of practical relevance is much more valuable than merely computing the currently available metrics. It also gives the reader an advantage to see the newly created data modalities in practice, even though the computational cost of evaluation is high. Nevertheless, we also provide the evaluation results using the standard metrics in Table S2.

To obtain the main results, we trained DF for 70 epochs on BENCHMARK, FAKET, and BASELINE training data, every time across 6 different seeds of randomness (3×6 times in total). Each model was tested on the same test tomogram – *model_9* from the original SHREC data set. The final comparison was done by observing the models' performances in terms of F1 score for localization task and F1 macro score for classification task. The computation of the scores was done the same way as in[7] where also further particularities of the tasks are described. To estimate the 68% and 95% confidence intervals, we used the bootstrapping method. The results are visualized and discussed in Results. Results of additional experiments are presented in the Supplementary material.

DeepFinder

DeepFinder emerged from the SHREC challenge as one of the most successful candidates. One other reason why we chose to use DF was the availability of its source code. DF is a deep 3D-convolutional neural network trained in a supervised fashion by optimizing a dice loss. For training of the models, we used the Adam optimizer with learning rate of 0.0001. The exponential decay rate was set to 0.9 for the first moment estimate and 0.999 for the second moment estimate as in.[9]

To train the DF, we used tomograms containing particle instances from 16 classes (two of those being background and vesicles). We adjusted the training procedure of DF such that in one epoch, all patches containing particles within the training data set were seen once, or with minimal repetition. a patch in a shape of a box is rotated by 180° at random. Patches are processed in batches constructed from a randomly permuted list of all particles in the training set. After our changes in the training procedure, an epoch represents a different number of gradient steps than in,[9] therefore the numbers of epochs are not comparable with the original paper. To perform the particle localization and -classification on the test tomogram, DF carries out a segmentation procedure followed by clustering and thresholding steps, where the two latter steps are computed using a CPU. The thresholding is used to reduce the number of false positive findings.

To train the DF network implemented in Keras framework[29] using TensorFlow backend,[30] we utilized multiple *NVIDIA A100 40GB SXM4* GPUs. That allowed us to run multiple experiments in parallel. One training job comprising of 9 training tomograms required approx. 17GB of GPU memory, therefore we were able to submit two jobs on one GPU at a time. One training epoch, as defined earlier, took $\approx$ 21 min. to finish. The segmentation of a test tomogram took $\approx$ 2.5 min. on the aforementioned GPU. The clustering step took $\approx$ 30 min. on a single core of a 2×64-*core AMD EPYC 7742* CPU. The most computational expensive step was thus the training of the DF.