ProPicker: Promptable Segmentation for Particle Picking in Cryogenic Electron Tomography

Simon Wiedemann[1], **Zalan Fabian**[2], **Mahdi Soltanolkotabi**[2], **Reinhard Heckel**[1]
[1] Department of Computer Engineering, Technical University of Munich
[2] Department of Electrical and Computer Engineering, University of Southern California

Abstract

Cryogenic electron tomography (cryo-ET) is a technique to produce highly detailed 3D images (called tomograms) of cellular environments. Cryo-ET is currently the only technique that can achieve near-atomic resolution of proteins and cellular structures in their native environment. An essential step of cryo-ET analysis techniques targeted at protein structure determination is to find all instances of the protein of interest in the tomograms, a task known as *particle picking*. Due to the low signal-to-noise ratio, presence of artifacts and vast diversity in target proteins, particle picking is a challenging 3D object detection problem. Existing approaches for particle picking are either slow or are limited to picking a few particles of interest, which requires large annotated and difficult to obtain training datasets. In this work, we propose ProPicker, a fast and universal particle picker that can detect particles beyond those included in the training set and can process tomograms within a few minutes. Our promptable design allows for selectively detecting a specific protein in the volume based on an input prompt. Our experiments demonstrate that ProPicker can achieve performance on par with state-of-the-art universal pickers, while being up to an order of magnitude faster.

1 Introduction

Cryo-electron tomography (cryo-ET) has recently surged in popularity due to its unique capabilities of imaging biological macromolecules in their native environments (Turk & Baumeister, 2020; Hylton & Swulius, 2021). An ambitious goal of cryo-ET is to obtain an 'atlas' of the cell with all of its constituent macromolecules mapped in their native environment. This would revolutionize our understanding of essential protein interactions and has the potential to provide breakthroughs in modern medicine spanning cell biology to drug discovery (Bodakuntla et al., 2023). In this paper, we focus on particle picking, an essential task in cryo-ET imaging and analysis, which entails finding all instances of a particle of interest in 3D volumes, called tomograms, obtained with cryo-ET.

Particle picking is a challenging 3D object detection problem. Due to the fundamental limitations of data acquisition in cryo-EM, tomograms have a very low signal-to-noise ratio and exhibit strong artifacts. Moreover, depending on the goal of the cryo-ET study, practitioners need to analyze substantial amounts of data. Tomograms are often large ($200 \times 1000 \times 1000$ voxels and up), and cryo-ET datasets can consist of more than a hundred tomograms (Genthe et al., 2023; Zeng et al., 2023). Finally, due to the significant diversity in protein types within the cell, there is a vast array of unique object classes to be detected, many of which display only subtle differences, rendering differentiation challenging. For instance, the human body alone is estimated to contain more than 20,000 unique proteins (Li & Buck, 2021). Given these challenges, a particle picking method should be accurate, fast, and universal, i.e. should be able to pick any particle of interest. Existing methods for particle picking are either slow or not universal, that is they are limited to picking a small, fixed set of particles of interest.

Machine Learning for Structural Biology Workshop, NeurIPS 2024.

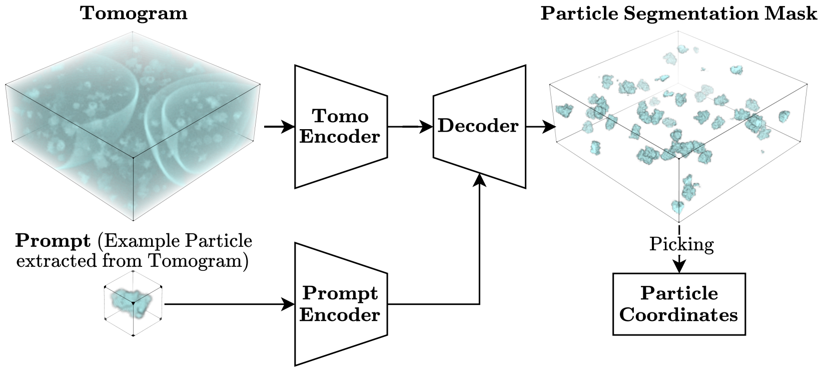
Tomogram
Particle Segmentation Mask
Tomo Encoder
Decoder
Prompt (Example Particle extracted from Tomogram)
Prompt Encoder
Picking
Particle Coordinates

Figure 1: Overview of ProPicker: We extract a generalized representation of the particle to be picked in the tomogram by a prompt encoder. Conditioned on the prompt features, we segment the tomogram for the desired particle. Finally, we leverage the output segmentation map to find the particle coordinates, either by clustering or template-based approaches.

In this paper, we propose ProPicker, a **Pro**mptable particle **Picker** that can rapidly detect any type of protein selected by a versatile prompting mechanism (Figure 1). For fast particle picking, ProPicker leverages an efficient 3D segmentation network to segment particles of interest in tomograms and accurately locate their positions. To make ProPicker universal, we propose a novel promptable segmentation architecture that uses a conditioning mechanism to control the type of particle to be segmented by the network. The prompt provides a generalized representation of the particle one wishes to pick and is not restricted to those encountered during training.

We demonstrate that ProPicker can accurately pick a wide array of unique proteins while simultaneously being significantly faster at single particle picking than the state-of-the-art (Section 4.1). We also demonstrate that ProPicker can pick new particles that the model has not encountered during training (Section 4.2. Experiments on two real-world tomograms indicate that ProPicker is able to pick particles in a crowded cellular context and in diverse environments (Section 4.3). Finally, we demonstrate that ProPicker can be fine-tuned with little data to improve picking performance on challenging novel particles (Appendix E).

2 Background & Related Work

The most widely used method for particle picking is template matching (TM) (Bohm et al., 2000; Cruz-León et al., 2024). TM methods pick particles by comparing a template of the particle to be picked to a candidate sub-tomogram extracted via a 3D sliding window. This approach is universal, as it can pick any particle as long as a template is available. However, TM is computationally demanding, as the stride of the sliding window needs to be small for accurate picking. TM can take up to hours per tomogram Genthe et al. (2023); Maurer et al. (2024).

Building upon classical template-based approaches, TomoTwin Rice et al. (2023) utilizes a learned convolutional encoder to map both template and sub-tomogram into a structured latent space, where similarity is evaluated. TomoTwin is state-of-the-art among universal particle pickers.

Particle pickers using deep-learning-based object detection often outperform TM in terms of performance and picking speed (Gubins et al., 2020; Genthe et al., 2023). Many such pickers, including our ProPicker method, use a convolutional network to segment particles of interest belonging to one or more classes and produce candidate particle locations by clustering the predicted segmentation masks (Moebel et al., 2021; De Teresa-Trueba et al., 2023; Liu et al., 2024). While deep-learning-based object detection approaches for particle picking are typically significantly faster than TM-like methods (Gubins et al., 2020), current variants are not universal, i.e., they can only pick a few fixed particles of interest seen in the training set.

ProPicker belongs to the class of particle pickers that use deep learning-based object detection. As such, ProPicker inherits the faster picking speed compared to template-based methods. What distinguishes ProPicker from existing pickers using deep learning-based object detection, is that due to its promptable design, ProPicker is universal, i.e., is not limited to a fixed set of particles. We note that the method that is most closely related to ProPicker is TomoTwin, which is also universal, but template-matching-like in nature.

3 ProPicker: Promptable Segmentation for Particle Picking

ProPicker takes a tomogram $\boldsymbol{x}$, and a prompt, i.e., a 3D voxel array $\boldsymbol{p} \in \mathbb{R}^{m \times m \times m}$ representing the particle of interest as input. The output is a set $\mathcal{C}$ containing predicted particle centers. In this work, a prompt $\boldsymbol{p}$ is a small sub-tomogram, i.e., a part of a larger tomogram, of shape $37 \times 37 \times 37$ containing one instance of the particle of interest. The prompt sub-tomogram is extracted from one of the tomograms to which we wish to apply particle picking. ProPicker consists of a prompt encoder and a segmentation model, which we detail in the following.

The prompt encoder. The prompt encoder $\varepsilon_{\boldsymbol{p}} : \mathbb{R}^{m \times m \times m} \to \mathbb{R}^d$ extracts a salient feature vector $\boldsymbol{z_p} \in \mathbb{R}^d$ that encodes information required for efficiently detecting the particle in tomograms. We focus on prompts in voxel-space and use the TomoTwin encoder (Rice et al., 2023) as the prompt encoder due to its robustness and good performance in template-based particle classification (see Section 2). Its input is a sub-tomogram that includes the particle of interest, and it outputs a concise ($d = 32$) representation $\boldsymbol{z_p}$ of the particle that we use to condition the segmentation model.

The segmentation model. Given an input volume $\boldsymbol{x} \in \mathbb{R}^{n \times n \times n}$ and prompt $\boldsymbol{p}$, our promptable segmentation model $\mathcal{S} : \mathbb{R}^{n \times n \times n} \times \mathbb{R}^d \to \mathbb{R}^{n \times n \times n}$ can be conditioned on the input prompt that steers the output map $\boldsymbol{y} \in \{0, 1\}^{n \times n \times n}$ to the desired particle class, that is $\boldsymbol{y} = \mathcal{S}(\boldsymbol{x}; \boldsymbol{z_p})$, where $\boldsymbol{z_p} = \varepsilon(\boldsymbol{p})$. The model output $\boldsymbol{y}$ is the voxel-wise prediction of the model with respect to the absence/presence of the particle described in the input prompt. We detail the architecture of the conditional segmentation model and how we train it in Appendix A and Appendix B.

Picking particles with ProPicker. The first step is to manually extract a sub-tomogram that includes an instance of the particle of interest to be used as a prompt. Next, we embed the extracted prompt and segment the tomogram using ProPicker. As tomograms are typically very large, we segment the volume using a strided 3D sliding window approach. Specifically, we slide a moderately sized window across the tomogram to extract sub-tomograms, and segment each sub-tomogram individually. We obtain a full-sized segmentation mask for the tomogram by combining the sub-tomogram level masks, averaging overlapping regions. We propose two strategies to map the segmentation output $\boldsymbol{y}$ to particle center coordinates $\mathcal{C}$:

- **Clustering-based picking (ProPicker-C):** We detect clusters in the segmentation map by finding connected components. The centroid of each cluster is a predicted particle center. The precision of this approach can be improved by leveraging prior information about the target particle size by excluding clusters that are too small or too large.
- **TM-based picking (ProPicker-TM):** We apply a TM-based picker to the input tomogram over regions where our segmentation mask predicts the presence of a particle.

4 Experiments

Here, we show that ProPicker can pick particles based on a single prompt and with high speed. We measure picking performance with F1 score, and report best-case performance with hyperparameters, e.g., particle-dependent thresholds on cluster sizes, optimized on test data for all methods following common practice (Rice et al., 2023).

Code for training and picking with ProPicker is publicly available; see Appendix F for details.

4.1 Picking Speed

Both TomoTwin and ProPicker process the tomogram in a 3D sliding window fashion. Therefore, inference time is cubically related to the window stride s. However, large strides (small overlap) often result in low detection performance. Here, we explore this trade-off. To quantify speed, we report the throughput in tomograms per hour on a single NVIDIA L40 GPU for picking a single particle of interest in a tomogram of size $200 \times 512 \times 512$.

As the speed at which a particle can be reliably picked depends on, e.g., the particle's size (especially for TM methods like TomoTwin), we measure the picking speed on a set of 10 tomograms which contain instances of 100 unique particle classes in total. Both TomoTwin ProPicker and have seen all of these particles during training, but within in different tomograms, i.e., in different contexts.

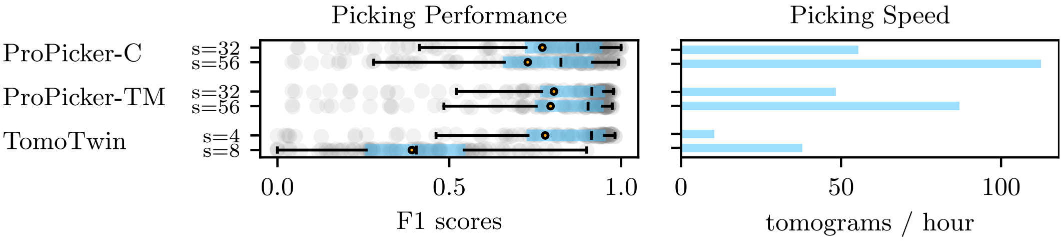

Figure 2: Best-case F1 scores and speed for ProPicker-C, ProPicker-TM (with TomoTwin TM) and TomoTwin for 100 unique particles seen during training. Vertical markers are medians, circles are means.

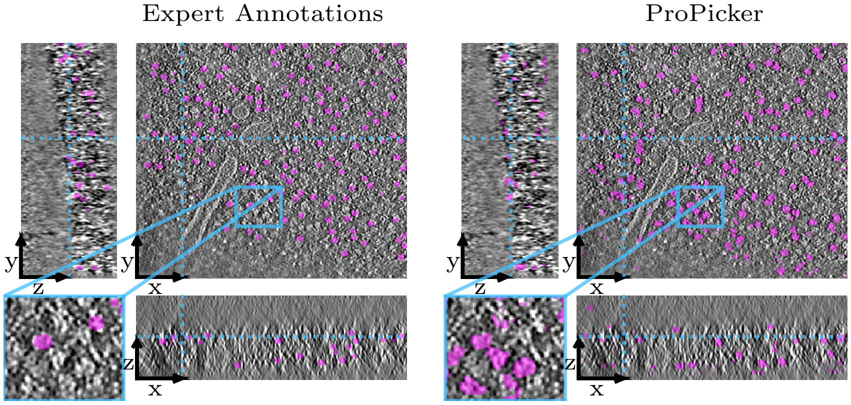

Figure 3: Slices through a tomogram (TS_30) from EMPIAR 10988. For clarity, we denoise the tomogram with Topaz (Bepler et al., 2020); the Topaz-denoised tomogram is not used for picking.

As can be seen in Figure 2, ProPicker-C with $s = 32$ can pick most particles as well as TomoTwin for $s = 4$ (TomoTwin's default $s = 2$ gives slightly better performance but is almost $8\times$ slower), while being more than $5\times$ faster. Increasing ProPicker's stride to $s = 56$ doubles the speed while resulting in a moderate loss of performance. Note that TomoTwin cannot be significantly accelerated by increasing the stride, as even $s = 8$ leads to a large drop in performance. ProPicker-TM with TomoTwin as template matching yields better average performance than ProPicker-C. This comes at the moderate cost of searching ProPicker's segmentation mask with TomoTwin (with $s = 4$).

4.2 Generalization to Unseen Particles in Synthetic Tomograms

Next, we study ProPicker's capability to generalize to unseen particles. We test the generalization capability of ProPicker on the TomoTwin generalization tomogram simulated by Rice et al. (2023). This tomogram contains instances of 8 structurally diverse particles that are not part of our training set. As it has been generated with the same simulator as most of ProPicker's and TomoTwin's training dataset, the tomogram is therefore well suited for studying the generalization of ProPicker to unseen particles in an environment similar to that seen during training. Across the 8 particle classes, ProPicker-C achieves a mean best-case F1 of 0.81 which is on par with TomoTwin's mean best-case F1 of 0.83. The detailed results can be found in Appendix C.

4.3 Generalization to Real-World Tomograms

As a first example, we consider a single tomogram from EMPIAR 10988, which shows ribosomes within *S. pombe* cells (De Teresa-Trueba et al., 2023). Figure 3 shows ProPicker ($s = 32$) segmentation masks without clustering or TM-based picking alongside expert annotations of the ribosomes produced by De Teresa-Trueba et al. (2023). For a quantitive evaluation, we compute best-case picking F1 scores with respect to the expert annotations. ProPicker-C achieves a best-case F1 score of 0.35, TomoTwin ($s = 4$) achieves 0.60. We found that ProPicker-C performs better when we slightly denoise the tomogram with a Gaussian filter with kernel standard deviation 0.5. On the denoised tomogram, ProPicker-C achieves a best-case F1 score of 0.46.

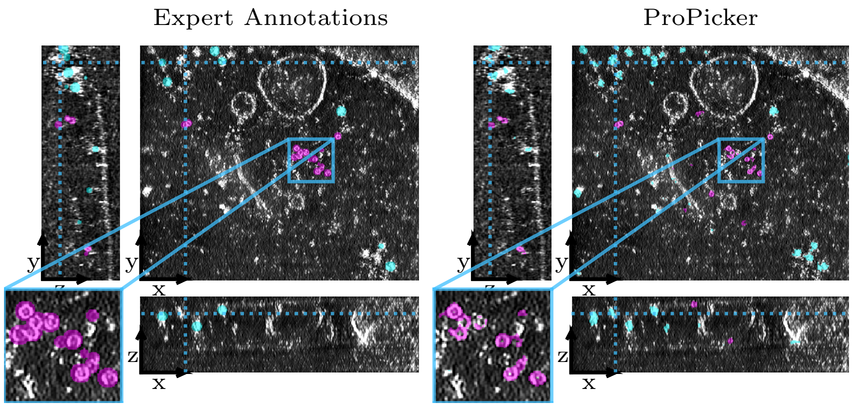

Figure 4: Slices through a tomogram from the DS-10440 dataset (Peck et al., 2024). Expert annotations and ProPicker segmentation masks for apoferritin and cytosolic ribosomes are shown in pink and ice-blue. The tomogram underlying the segmentation masks has been denoised by Peck et al. (2024). We use the raw, un-denoised tomogram (not shown) as input for ProPicker.

Next, we consider a single tomogram from the DS-10440 dataset generated by Peck et al. (2024), which is available through the Cryo ET Data Portal (Ermel et al., 2024). Among other particles (see Appendix D), the dataset contains expert annotations for cytosolic ribosomes (4.3 megadalton) and apoferritin (450 kilodalton). Our goal is to pick all instances of the ribosomes and the apoferritin in the same tomogram. As can be seen in Figure 4, the segmentation model of ProPicker is able to detect both the ribosomes (ice blue) and the apoferritin (pink) based on a single prompt each. Using ProPicker-TM with TomoTwin as template matching, we obtained a best-case F1 picking score of 0.69 for the ribosomes and 0.71 for the apoferritin. The cluster-based picking approach of ProPicker-C gave a best-case F1 picking score of 0.5 and 0.53 respectively. The worse performance of cluster-based picking is likely due to crowded parts (see, e.g. the zoomed-in region), where several instances of the same particle appear close together, which makes clustering challenging compared to peak-finding, which is used in TomoTwin-based picking. TomoTwin applied to the entire tomogram achieves a best-case F1 score of 0.64 for both the ribosomes and the apoferritin.

5 Discussion & Conclusion

In this work, we propose ProPicker, a particle picking method for cryo-ET that leverages a novel promptable segmentation model for rapid and accurate detection of proteins. The core of our framework is an efficient segmentation model capable of selectively detecting particles in tomograms based on an input prompt, a concise representation of the particle of interest. Through the use of a 3D segmentation U-Net, ProPicker greatly accelerates particle picking while matching state-of-the-art performance on a wide range of particles. Our experiments show that ProPicker can generalize to particles unseen during training and also to real-world tomograms, although ProPicker's training set consists exclusively of synthetic data (see Appendix B).

We also encountered real-world tomograms and particles for which ProPicker was unable to produce good results (see Appendix D, Appendix E). Such issues related to generalization and robustness are not exclusive to ProPicker (Bandyopadhyay et al., 2022), e.g., Huang et al. (2024) reported significant drops in performance when applying deep learning-based particle pickers, among them TomoTwin, to tomograms whose characteristics are too different from the training data. One approach to improve the performance of ProPicker in challenging scenarios is fine-tuning (see Appendix E).

It is widely accepted that training on larger and more diverse datasets improves the robustness of deep learning models (Radford et al., 2021; Fang et al., 2022; Lin & Heckel, 2024). Therefore, a promising direction for future work is to collect large datasets of tomograms with ground truth particle annotations for a variety of particles. Large scale efforts to do so have already been initiated, see for example the CryoET Data Portal (Ermel et al., 2024) and (Ishemgulova et al., 2023). Once such datasets become available, incorporating them into the training sets of universal particle pickers like TomoTwin and ProPicker is likely to increase their robustness and performance.

References

H. Bandyopadhyay, Z. Deng, L. Ding, S. Liu, M. R. Uddin, X. Zeng, S. Behpour, and M. Xu. Cryo-shift: reducing domain shift in cryo-electron subtomograms with unsupervised domain adaptation and randomization. *Bioinformatics*, 38(4):977–984, 2022.

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, 2020.

S. Bodakuntla, C. C. Kuhn, C. Biertümpfel, and N. Mizuno. Cryo-electron microscopy in the fight against covid-19—mechanism of virus entry. *Frontiers in Molecular Biosciences*, 10:1252529, 2023.

J. Bohm, A. S. Frangakis, R. Hegerl, S. Nickell, D. Typke, and W. Baumeister. Toward detecting and identifying macromolecules in a cellular context: template matching applied to electron tomograms. *Proceedings of the National Academy of Sciences*, 97:14245–14250, 2000.

S. Cruz-León, T. Majtner, P. C. Hoffmann, J. P. Kreysing, S. Kehl, M. W. Tuijtel, S. L. Schaefer, K. Geißler, M. Beck, B. Turoňová, and G. Hummer. High-confidence 3d template matching for cryo-electron tomography. *Nature Communications*, 15(1):3992, 2024.

I. De Teresa-Trueba, S. K. Goetz, A. Mattausch, F. Stojanovska, C. E. Zimmerli, M. Toro-Nahuelpan, D. W. C. Cheng, F. Tollervey, C. Pape, M. Beck, A. Diz-Muñoz, A. Kreshuk, J. Mahamid, and J. B. Zaugg. Convolutional networks for supervised mining of molecular patterns within cellular context. *Nature Methods*, 20(2):284–294, 2023.

Utz Ermel, Anchi Cheng, Jun Xi Ni, Jessica Gadling, Manasa Venkatakrishnan, Kira Evans, Jeremy Asuncion, Andrew Sweet, Janeece Pourroy, Zun Shi Wang, et al. A data portal for providing standardized annotations for cryo-electron tomography. *Nature Methods*, pp. 1–3, 2024.

A. Fang, G. Ilharco, M. Wortsman, Y. Wan, V. Shankar, A. Dave, and L. Schmidt. Data determines distributional robustness in contrastive language image pre-training (clip). In *International Conference on Machine Learning*, pp. 6216–6234, 2022.

E. Genthe, S. Miletic, I. Tekkali, R. Hennell J., T. C. Marlovits, and P. Heuser. Pickyolo: Fast deep learning particle detector for annotation of cryo electron tomograms. *Journal of Structural Biology*, 215(3):107990, 2023.

I. Gubins, M. L. Chaillet, G. van Der Schot, R. C. Veltkamp, F. Förster, Y. Hao, X. Wan, X. Cui, F. Zhang, E. Moebel, et al. Shrec 2020: Classification in cryo-electron tomograms. *Computers & Graphics*, 91:279–289, 2020.

Q. Huang, Y. Zhou, and A. Bartesaghi. Milopyp: self-supervised molecular pattern mining and particle localization in situ. *Nature Methods*, pp. 1–10, 2024.

Qinwen Huang, Ye Zhou, Hsuan-Fu Liu, and Alberto Bartesaghi. Accurate detection of proteins in cryo-electron tomograms from sparse labels. In *European Conference on Computer Vision*, pp. 644–660. Springer, 2022.

R. K. Hylton and M. T. Swulius. Challenges and triumphs in cryo-electron tomography. *iScience*, 24(9):102959, 2021.

A. Ishemgulova, A. J. Noble, T. Bepler, and A. De Marco. Preparation of labeled cryo-et datasets for training and evaluation of machine learning models, 2023.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Z.-L. Li and M. Buck. Beyond history and "on a roll": The list of the most well-studied human protein structures and overall trends in the protein data bank. *Protein Science*, 30(4):745–760, 2021.

K. Lin and R. Heckel. Robustness of deep learning for accelerated MRI: Benefits of diverse training data. In *41st International Conference on Machine Learning*, volume 235, pp. 30018–30041, 2024.

G. Liu, T. Niu, M. Qiu, Y. Zhu, F. Sun, and G. Yang. Deepetpicker: Fast and accurate 3d particle picking for cryo-electron tomography using weakly supervised deep learning. *Nature Communications*, 15(1):2090, 2024.

V. J. Maurer, M. Siggel, and J. Kosinski. Pytme (python template matching engine): A fast, flexible, and multi-purpose template matching library for cryogenic electron microscopy data. *SoftwareX*, 25:101636, 2024.

E. Moebel, A. Martinez-Sanchez, L. Lamm, R. D. Righetto, W. Wietrzynski, S. Albert, D. Larivière, E. Fourmentin, S. Pfeffer, J. Ortiz, et al. Deep learning improves macromolecule identification in 3d cellular cryo-electron tomograms. *Nature Methods*, 18(11):1386–1394, 2021.

Ariana Peck, Yue Yu, Jonathan Schwartz, Anchi Cheng, Utz Heinrich Ermel, Saugat Kandel, Dari Kimanius, Elizabeth Montabana, Daniel Serwas, Hannah Siems, et al. Annotating cryoet volumes: A machine learning challenge. *bioRxiv*, pp. 2024–11, 2024.

E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville. Film: Visual reasoning with a general conditioning layer. *AAAI Conference on Artificial Intelligence*, 32(11), 2018.

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763, 2021.

G. Rice, T. Wagner, M. Stabrin, and S. Raunser. Tomotwin demonstration dataset (1.0), 2022. URL `https://doi.org/10.5281/zenodo.7186070`.

G. Rice, T. Wagner, M. Stabrin, O. Sitsel, D. Prumbaum, and S. Raunser. Tomotwin: generalized 3d localization of macromolecules in cryo-electron tomograms with structural data mining. *Nature Methods*, 20(66):871–880, 2023.

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, 2015.

M. Turk and W. Baumeister. The promise and the challenges of cryo-electron tomography. *FEBS Letters*, 594(20):3243–3261, 2020.

X. Zeng, A. Kahng, Liang Xue, J. Mahamid, Y.-W. Chang, and M. Xu. High-throughput cryo-et structural pattern mining by unsupervised deep iterative subtomogram clustering. *Proceedings of the National Academy of Sciences*, 120(15):e2213149120, 2023.

Appendix

A Architectural Details

Here, we provide details on our concrete choice for the segmentation model, and how we condition it on a prompt.

A.1 Segmentation Model Architecture

We use a well-established convolutional 3D U-Net (Ronneberger et al., 2015), which is an encoder-decoder architecture (see Figure 1), as our segmentation model. The U-Net's encoder consists of 5 spatial downsampling layers, and the corresponding decoder has 5 spatial upsampling layers. In total, the U-Net has 124 million trainable parameters.

A.2 Prompt Conditioning Technique

We condition each of the decoder's 5 spatial upsampling layers with FiLM (Perez et al., 2018), which works as follows: Let C be the number of channels (features) of an intermediate 3D feature map after upsampling. First, we multiply the encoded prompt $\boldsymbol{z_p} \in \mathbb{R}^{32}$ with two (learnable) matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{C \times 32}$. Finally, we map each channel $k \in \{1, ..., C\}$, with an affine transformation with slope $(\boldsymbol{A}\boldsymbol{z_p})_k \in \mathbb{R}$ and intercept $(\boldsymbol{B}\boldsymbol{z_p})_k \in \mathbb{R}$, which gives the conditioned feature map. We use one separate pair of matrices $(\boldsymbol{A}, \boldsymbol{B})$ for each of the 5 upsampling layers.

B Training details

Training Dataset of ProPicker. We train ProPicker on realistically simulated tomograms from Rice et al. (2023) and Gubins et al. (2020), which have also been used to train TomoTwin. Our training set contains the majority of TomoTwin's training data, and consists of 78 tomograms containing a total of 113 unique protein types, as well as gold fiducial markers, vesicles and filaments. Each tomogram contains around 1500 protein instances, each belonging to a set of $10 - 13$ unique protein types. We train on sub-tomograms of size $64 \times 64 \times 64$ extracted from all tomograms with a 3D sliding window with stride 32.

Training ProPicker. To reduce computational cost, we keep the prompt encoder frozen during training. We train the segmentation model with the Adam optimizer (Kingma & Ba, 2015) with fixed learning rate 0.01. For each gradient step, we first randomly sample a batch of 8 sub-tomograms. Each sub-tomogram can contain particles that belong to up to 13 unique classes (see above). Next, we randomly sample 10 prompts. Each prompt corresponds to one particle class of which instances may be contained in the sub-tomogram. We pass each sub-tomogram and its corresponding 10 prompts through the conditional segmentation model. This yields a total of $80 = 8 \cdot 10$ predicted segmentation masks. Finally, we compute the average voxel-wise binary cross entropy between the model outputs and the 80 single-class target masks as loss.

C Performance of TomoTwin and ProPicker-C on the TomoTwin generalization Tomogram

Method / PDB	1avo	1e9r	1fpy	1fzg	1jz8	1oao	2df7	Mean	Median
TomoTwin	0.71	**0.80**	**0.70**	0.90	0.85	**0.86**	0.99	**0.83**	**0.85**
ProPicker-C	**0.79**	0.79	0.56	**0.94**	**0.87**	0.71	1.00	0.81	0.79

Table 1: Best-case F1 picking scores for TomoTwin (s = 2) and ProPicker-C (s = 32) on the TomoTwin generalization tomogram, which contains particles that were not seen during training. We took TomoTwin's results from the authors' software demo (Rice et al., 2022).

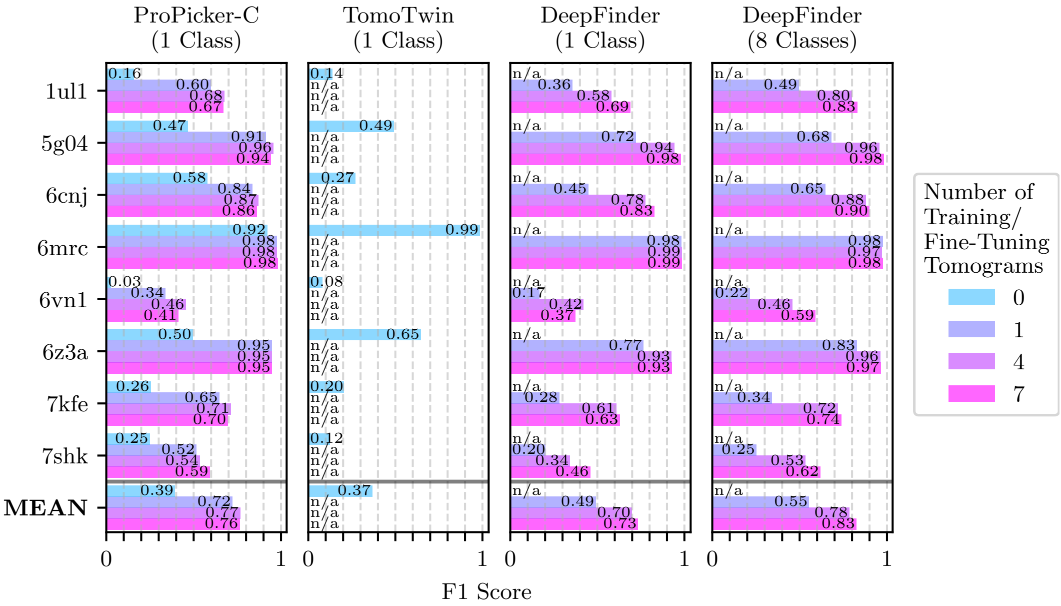

Figure 5: Best-case F1 scores of ProPicker-C and DeepFinder versus amount of fine-tuning data for unseen particles.

D Limitations and Challenges

In the main paper, we demonstrated that ProPicker generalizes well to unseen particles and real-world tomograms. However, certain tomograms and particle types posed challenges, highlighting opportunities for further improvement.

For example, the tomogram by Peck et al. (2024) in which we picked ribosomes and apoferritin (Section 4.3) also contains virus-like particles, beta amylase, beta galactosidase, and thyroglobulin. In our hands, neither ProPicker nor TomoTwin were able to pick these particles based on a single prompt. Beta Amylase, beta galactosidase, and thyroglobulin are very small, produce little contrast and are, therefore, considered very hard to pick (Peck et al., 2024). The virus-like particles in contrast are large and clearly visible even in the noisy tomograms. However, the training dataset of ProPicker and TomoTwin contains only proteins and no virus-like particles (Rice et al., 2023), which might explain the low performance.

In Appendix E, we show results on a synthetic tomogram where ProPicker achieves mixed picking performance on 8 unique particle classes. We also show that it is possible to significantly improve ProPicker's performance through fine-tuning on a comparably small amount of data.

E Fine-Tuning ProPicker

In the main paper, we have shown that ProPicker is able to accurately pick particles both seen and unseen during training based on a single prompt. However, we have also seen that, especially in the real-world tomograms, there are some particles which neither ProPicker nor TomoTwin are able to locate. Here, we show that ProPicker's performance on unseen particles can be significantly, and data-efficiently improved through fine-tuning. Note that fine-tuning ProPicker is different from the setup of picking based on a single prompt we have considered so far in this work, but is analogous to the approach of non-universal picking methods. Rather than trying to outperform such methods, our aim is to show that fine-tuning ProPicker-C can perform on-par with them.

Fine-tuning strategy for ProPicker. We describe how to fine-tune ProPicker to pick a single particle of interest. This requires (parts of) one or more tomograms with corresponding ground-truth binary segmentation masks of the particle, as well as a manually extracted example of the particle that serves as prompt. During fine-tuninig, we keep this prompt and the prompt encoder fixed, and only fine-tune the segmentation model and the prompt conditioning mechanism.

Dataset. We resort to a set of tomograms from TomoTwin's training set each of which contains instances of 8 unique particles. As TomoTwin's training set was specifically desinged to contain particles whose structures are all very different from one another (see (Rice et al., 2023) for details), these 8 particles are hard, unseen examples for ProPicker. We have access to 8 tomograms, out of which we use 7 for fine-tuning and one for testing. All tomograms contain around 150 instances of each particle. For the experiment in this Section, we re-trained ProPicker and its TomoTwin prompt-encoder, and excluded the 8 particles from the both training sets.

DeepFinder baseline. We choose DeepFinder (Moebel et al., 2021) as baseline from the class of non-universal state-of-the-art deep learning-based particle pickers. DeepFinder is segmentation-based and uses a 3D U-Net-like convolutional architecture, like many other such pickers, e.g, DeePiCt (De Teresa-Trueba et al., 2023) or DeepETPicker (Liu et al., 2024).

We compare to two variants of DeepFinder. For the first variant, "DeepFinder (1 Class)", we train one DeepFinder model for each particle separately. This is the same setup as when we fine-tune ProPicker. For the second variant, "DeepFinder (8 Classes)", we train one DeepFinder model to pick all 8 particles *simultaneously*. Moebel et al. (2021) observed that multi-class training can yield substantial improvements in performance for hard-to-pick particles. Note that, in contrast to the single class setup, the multi-class setup requires annotations for *all 8* particles.

Results. Due to the particularly challenging data, we observe rather low picking F1 scores for ProPicker-C on most particles when picking with a single prompt (left panel of Figure 5). Note that the (re-trained) TomoTwin, too, struggles with picking, and achieves a mean F1 score similar to ProPicker-C. As it is not straightforward how to fine-tune TomoTwin on individual particles, we only report the performance of the re-trained TomoTwin for prompt-based picking.

Fine-tuning ProPicker-C significantly boosts picking performance for all particles. The performance of the fine-tuned models saturates quickly as more data becomes available: Fine-tuning on a single tomogram, which contains around 150 instances of each particle, yields significant improvements for all particles, whereas going from 4 to 7 fine-tuning tomograms makes little to no difference.

If only a few tomograms *or* only annotations for the particle of interest are available, fine-tuning ProPicker-C yields superior performance compared to both variants of DeepFinder. When training/fine-tuning on a single tomogram, the fine-tuned ProPicker-C pickers outperform DeepFinder with 1 class (center panel) and 8 classes (right panel). Even as more data becomes available, ProPicker-C performs as well as or better than the single-class DeepFinder models, but the performance gap is narrowing.

The benefit of ProPicker-C's pre-training is not able to outweigh the advantages of multi-class particle picking if enough training data is available: DeepFinder trained on all 8 classes simutaneously performs on par or better than the 8 fine-tuned ProPicker-C pickers when training/fine-tuning on 4 or 7 tomograms. We again emphasize that the price of the improved performance is having to annotate *all* particles in the tomogram even if one is only interested in a single one.

Further reducing the data requirements for fine-tuning ProPicker by incorporating techniques for data-efficient training (see (Huang et al., 2022)) is a promising direction for future work.

F Code Availability

Python scripts for training and picking with ProPicker can be downloaded here: `https://drive.google.com/file/d/1h6fJFlWZOhfJfL33yd4Z4yBfM_VlfnKx/view?usp=sharing`. Please refer to the `README` file for details.

A pre-trained model is avaialble here: `https://drive.google.com/file/d/1QQcpvFzgMZcWF_Yuba8O3tJJ4pmpu6EB/view?usp=share_link`.