


UPicker: a semi-supervised particle picking transformer method for cryo-EM micrographs

Chi Zhang¹ , Yiran Cheng², Kaiwen Feng¹, Fa Zhang³, Renmin Han^{2*}, Jieqing Feng^{1*}

¹State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou 310058, Zhejiang, China

²Research Center for Mathematics and Interdisciplinary Sciences, Shandong University, Qingdao 266000, Shandong, China

³School of Medical Technology, Beijing Institute of Technology, Beijing 100081, China

*Corresponding authors. Jieqing Feng, State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou 310058, Zhejiang, China. E-mail: jqfeng@cad.zju.edu.cn;

Renmin Han, Research Center for Mathematics and Interdisciplinary Sciences, Shandong University, Qingdao 266000, Shandong, China.

E-mail: hanrenmin@gmail.com

Abstract

Automatic single particle picking is a critical step in the data processing pipeline of cryo-electron microscopy structure reconstruction. In recent years, several deep learning-based algorithms have been developed, demonstrating their potential to solve this challenge. However, current methods highly depend on manually labeled training data, which is labor-intensive and prone to biases especially for high-noise and low-contrast micrographs, resulting in suboptimal precision and recall. To address these problems, we propose UPicker, a semi-supervised transformer-based particle-picking method with a two-stage training process: unsupervised pretraining and supervised fine-tuning. During the unsupervised pretraining, an Adaptive Laplacian of Gaussian region proposal generator is proposed to obtain pseudo-labels from unlabeled data for initial feature learning. For the supervised fine-tuning, UPicker only needs a small amount of labeled data to achieve high accuracy in particle picking. To further enhance model performance, UPicker employs a contrastive denoising training strategy to reduce redundant detections and accelerate convergence, along with a hybrid data augmentation strategy to deal with limited labeled data. Comprehensive experiments on both simulated and experimental datasets demonstrate that UPicker outperforms state-of-the-art particle-picking methods in terms of accuracy and robustness while requiring fewer labeled data than other transformer-based models. Furthermore, ablation studies demonstrate the effectiveness and necessity of each component of UPicker. The source code and data are available at <https://github.com/JachyLikeCoding/UPicker>.

Keywords: cryo-EM; particle picking; object detection; unsupervised pretraining; transformer

Introduction

Cryo-electron microscopy (cryo-EM) is a powerful three-dimensional (3D) bioimaging technique for visualizing biological macromolecules at near-atomic resolution without requiring crystallization and is widely applied in structural biology [1]. One of its key applications is single-particle analysis (SPA), where 3D structures are reconstructed from a series of 2D micrographs captured by an electron microscope. Accurate particle picking is essential for reconstructing high-resolution 3D structures. It is also a challenge due to the low signal-to-noise ratio (SNR) micrographs. It involves detecting and extracting individual but randomly oriented macromolecular particles while avoiding misselection of contaminants, carbon film regions, and malformed particles [2]. It is usually necessary to select more than 100 000 particles for a near-atomic structure reconstruction [3]. Advancements in cryo-EM data acquisition technologies have substantially increased the demand for accurate, robust, and efficient particle-picking methods.

Traditional automatic particle-picking methods, such as XMIPP [4], DoG Picker [5], FindEM [6], EMAN2 [7], APPION [8], and RELION [9], are mainly based on template matching or specific feature extraction techniques. However, they are often error-prone,

require manual post-processing, and show reduced performance when dealing with non-ideal datasets characterized by high noise, low contrast, heterogeneous particle populations, or significant ice contamination.

In recent years, deep learning-based methods, particularly those leveraging convolutional neural networks (CNNs), have emerged as powerful tools for particle-picking in cryo-EM. Notable examples include DeepPicker [10], FastParticlePicker [11], AutoCryoPicker [12], crYOLO [13], Topaz [14], Warp [15], and CenterPicker [16]. These approaches typically involve training a neural network on a manually selected subset of particles, and then automatically picking particles across the dataset using the trained model. Despite their advancements, these methods still require significant amounts of labeled data, often fail to achieve satisfactory accuracy, and rely on complex post-processing and extensive parameter tuning.

Afterward, transformer-based models have emerged as a promising alternative, particularly because of their ability to capture global dependencies and model complex spatial relationships. The Detection with Transformer (DETR) model [17] has brought forth a new era of object detection by directly modeling object relations through self-attention mechanisms,

Received: September 9, 2024. Revised: October 31, 2024. Accepted: November 24, 2024

© The Author(s) 2024. Published by Oxford University Press.

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<https://creativecommons.org/licenses/by-nc/4.0/>), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited.

For commercial re-use, please contact journals.permissions@oup.com

providing a more holistic view of the image count. In the context of cryo-EM, this capability is crucial, as particles often exhibit subtle variations in morphology and are densely packed. Transformer-based methods like TransPicker [18] and CryoTransformer [19] have demonstrated that transformers can capture richer global information and be less sensitive to noise and carbon film regions. However, these models typically require more labeled data and longer training times compared with the CNN-based methods.

With limited training data, pretrained models become a valid option, gaining increasing popularity. These models are first trained on large-scale datasets and subsequently fine-tuned for specific tasks, reducing the need for extensive labeled data [20]. In recent years, self-supervised pretraining (PT) has shown promise in minimizing labeling costs while providing superior performance compared with traditional supervised PT methods [21]. In particle-picking, methods such as CrYOLO [13] and Topaz [14] implement PT strategies where a global model is initially trained on diverse datasets and then fine-tuned on specific datasets. EPicker [22] adopts a continuous learning approach by sequentially training on multiple datasets and fine-tuning (FT) on new data to update the model incrementally. CryoSegNet [23] leverages advanced pretrained foundational AI segmentation model for cryo-EM particle-picking. However, these methods rely heavily on labeled training data from extensive datasets and may underperform when applied to new data. Moreover, current PT methods often do not fully utilize the information available in the abundant unlabeled images from new datasets, suggesting potential for improvement in representation learning.

In this study, we propose a novel method named UPicker (particle PICKing transformER with Unsupervised Pretraining) for cryo-EM images. UPicker is designed to address the limitations of existing methods by leveraging the strengths of transformer models and self-supervised learning. UPicker initially undergoes unsupervised PT on unlabeled images to significantly reduce the reliance on labeled datasets, and then it is fine-tuned with a small subset of labeled images. A novel Adaptive-LoG region proposal generator produces high-recall pseudo-labels during PT, effectively guiding the model to learn critical features from the data. To handle densely packed and morphologically similar particles common in cryo-EM datasets, UPicker employs a contrastive denoising training (CDN) strategy that accelerates convergence and stabilizes bipartite matching throughout the two-stage training process. This strategy further enhances the model's robustness and ensures precise localization of particles. Additionally, a hybrid data augmentation strategy is employed, further enhancing UPicker's performance in scenarios with extremely limited labeled data.

Extensive experiments on both simulated and experimental real-world datasets demonstrate that UPicker outperforms existing state-of-the-art methods in detection accuracy. UPicker consistently shows robust performance across diverse datasets, effectively handling various particle sizes and shapes. It achieves near real-time inference speeds and reduces the reliance on extensive labeled data compared with other transformer-based methods while eliminating the need for complex post-processing steps. These advantages make UPicker a highly effective and adaptable solution for cryo-EM particle-picking challenges.

Methods

UPicker workflow overview

UPicker is designed to enhance particle-picking in cryo-EM micrographs through a four-stage workflow: preprocessing (PP), PT, FT, and picking (Fig. 1a).

Preprocessing. This stage aims to reduce noise, improve particle visibility, and optimize the inputs for subsequent processes. Micrographs undergo motion correction and frame averaging before PP. The micrographs are normalized and denoised using a bilateral filter. Image contrast is then enhanced by histogram equalization or contrast-limited adaptive histogram equalization. Furthermore, two optional operations can be adopted: (i) down-sampling large-scale images to speed up computation and reduce memory consumption, and (ii) splitting images into smaller, overlapping patches that are processed individually and later merged for final predictions. This patch-based approach is especially beneficial for micrographs with high particle densities (e.g. more than 300 particles), addressing challenges posed by detection transformers when handling numerous objects.

Pretraining. For the PT stage, UPicker employs approximately 100–300 unlabeled and preprocessed micrographs (U') and generates region proposals that serve as pseudo-labels, allowing the model to learn from patterns in data without explicit labels. This unsupervised learning stage equips the model with a basic understanding of particle features and distributions in cryo-EM micrographs.

Fine-tuning. The model is fine-tuned using a smaller set of labeled images (L'), typically consisting of 20–50 images labeled with particle coordinates and diameters. The number of required labeled images is significantly lower than TransPicker [18], which typically needs over 100 labeled images, and CryoTransformer [19], which relies on a sufficiently large labeled dataset. UPicker further reduces the labeling requirements by allowing partial labels in each image. Although the labels are limited in L', they provide critical information that enhances the model's accuracy.

Picking. After PT and FT, the model picks particles across the entire dataset. Due to the design of UPicker and its training strategies, the majority of redundant detections are effectively avoided. As an optional refinement step, non-maximum suppression (NMS) can be applied when necessary to filter overlapping detections, ensuring only the most confident results are retained. Finally, coordinate files are generated for each micrograph and used to extract the corresponding particles, which are then organized into stacks for downstream SPA.

UPicker model architecture

The UPicker model consists of four main components: an Adaptive Laplacian of Gaussian (A-LoG) region proposal generator, a CNN backbone, a multilayer Transformer encoder-decoder, and prediction heads.

The A-LoG region proposal generator analyzes the preprocessed, unlabeled micrographs to generate high-confidence region proposals that are likely to contain particles (Fig. 1b). These regions are used as pseudo-labels in the PT stage, enabling the model to learn meaningful features from data without manual labels. This component is crucial for overcoming the challenge of limited labeled data in cryo-EM.

The CNN backbone (ResNet50 [24] by default) extracts multiscale feature maps from preprocessed images, capturing both low- and high-level particle features at various resolutions. The multiscale feature extraction is essential for handling the inherent heterogeneity and different sizes of particles in cryo-EM micrographs. While CNNs effectively capture localized features, they are limited in their ability to capture long-range dependencies and global context, which is crucial for distinguishing particles from background noise and artifacts.

To address this limitation, the multilayer Transformer encoder-decoder architecture (Fig. 1c) is employed. The extracted multiscale features along with the corresponding positional

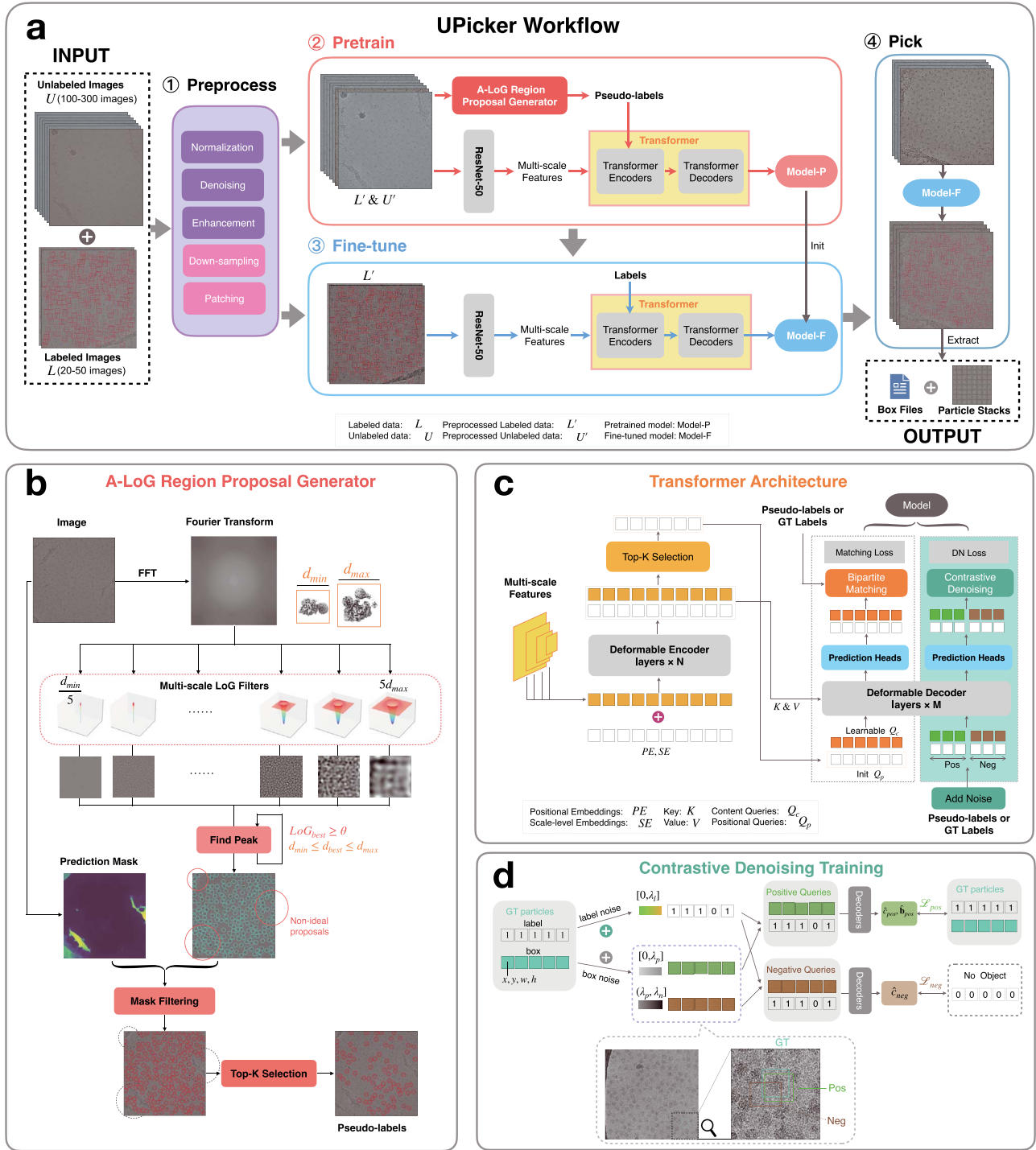


Figure 1. (a) UPicker identifies and localizes particles by a four-stage workflow: preprocessing, pretraining, fine-tuning, and picking. (b) Illustration of the Adaptive-LoG region proposal generator. (c) Illustration of the Transformer architecture used in UPicker. (d) The process of generating positive and negative queries and a demonstration of contrastive denoising training.

and scale-level embeddings are flattened and passed through the Transformer encoders for further refinement and feature enhancement. By utilizing the attention mechanism, the Transformer effectively integrates both local and global information, allowing it to adaptively focus on relevant regions, even in noisy micrographs where particles may be difficult to differentiate based solely on local features. The Transformer then employs a mixed query selection strategy: it initializes

high-quality positional queries (Q_p) from the encoder's output while maintaining content queries (Q_c) as learnable parameters in the decoders. This strategy enables the model to dynamically focus on relevant regions, and enhance the robustness of particle-picking. To efficiently handle a large number of particles in high-density micrographs, UPicker employs a multiscale deformable attention module [25]. This module combines the features output by the encoder and iteratively updates the queries at each decoder

layer, effectively modeling complex spatial dependencies while maintaining computational efficiency.

The prediction heads output both bounding boxes and classification scores for each detected region. A set-based loss is employed using the Hungarian algorithm [26] for bipartite matching to ensure a unique prediction for each ground truth (or pseudo-ground truth) bounding box. To further stabilize this matching process and accelerate convergence, UPicker adopts a CDN strategy [27].

Unsupervised pretraining with A-LoG region proposals

To alleviate the pressure of labeling large training sets, unsupervised PT based on region proposals is integrated to leverage unlabeled data. Region proposal methods for object detection have been extensively studied, with some representative methods including objectness [28], selective search [29], EdgeBox [30], and RPN [31]. Although these methods have shown promising performance in PT multiple object detection networks, they may not be suitable for obtaining candidate regions in cryo-EM micrographs. For instance, objectness methods typically rely on the texture, color, and shape features of objects. Selective search segments and merges super-pixels based on similarity to generate proposals. The EdgeBox method relies on edge information and does not perform satisfactorily in detecting multiple objects in a single category. Overall, the extremely low SNR, unclear edges, and dense particle distribution in cryo-EM images make it challenging for existing methods to accurately identify region proposals.

Before the rise of deep-learning methods, a common class of traditional template-free automatic particle-picking algorithms was filter-based algorithms, including the Laplacian of Gaussian (LoG) picker in RELION [32], the blob picker in CryoSPARC [33], etc. However, their detection is often affected by contaminated areas and carbon film areas, resulting in reduced accuracy. Considering all these factors, a novel A-LoG region proposal generator is proposed to obtain pseudo-labels for unsupervised PT.

A-LoG region proposal generator

The A-LoG region proposal generator is designed to detect potential particle locations through a three-step process: multiscale LoG filtering, mask filtering, and top-K selection (Fig. 1b). The process begins by applying a series of multiscale LoG filters to the micrographs in the Fourier domain. These filters are designed to detect spots within the image, typically representing particle locations. The LoG filter with an optimal response to particles of diameter d is defined as follows:

$$\text{LoG}(k) = \frac{|k|^2}{\sigma^2} \exp\left(-\frac{|k|^2}{2\sigma^2}\right) \quad (1)$$

where k denotes the image frequency, and $\sigma = 2/d$. The generator utilizes a series of multiscale LoG filters, each tailored to detect particles of a specific size. Given the estimated minimum (d_{\min}) and maximum (d_{\max}) particle diameter range, the generator employs 11 filters: four for detecting blobs smaller than d_{\min} , four for detecting blobs larger than d_{\max} , and three for detecting blobs within the specified size range.

Pixels are selected for potential particle identification based on two criteria: they must exhibit the highest LoG-filtered value (LoG_{best}) across all filtered micrographs, surpassing a predefined threshold θ . Additionally, the corresponding blob size (d_{best}) must fall within the particle diameter range. Particles are selected iteratively based on the remaining peak values, and the pixels within

a circle of diameter d_{best} around each selected particle are set to zero. This iteration continues until no suitable pixels remain. To standardize the threshold across datasets, the default threshold is set to the average of all LoG_{best} values. The picking threshold can be adjusted to control the number of particles selected.

However, the initial candidate regions often include areas with high-contrast ice contamination or carbon film, which are undesirable for accurate particle picking. To reduce such non-ideal regions, a mask-filtering step is applied using the Micrograph-Cleaner software package [34]. This software generates a predicted mask for each micrograph, which is then thresholded to classify the image into ideal and non-ideal regions. Region proposals located within non-ideal areas are filtered out. In the final step, to reduce computational cost and further improve recall, the top-K proposals are selected, represented as a set of bounding boxes $\{\mathbf{b}_i\}_{i=1}^K$, where $\mathbf{b}_i \in \mathbb{R}^4$.

Pretraining process

During the PT stage, UPicker is trained using the top-K region proposals generated by the A-LoG region proposal generator. The model aims to align its N outputs with these K proposals.

UPicker consists of two prediction heads: $H_{\text{box}}(\cdot)$ for predicting bounding boxes $\hat{\mathbf{b}}_i$, and $H_{\text{obj}}(\cdot)$ for predicting whether a given box represents a particle (\hat{c}_i). The model output is defined as $\hat{\mathbf{y}}_i = (\hat{\mathbf{b}}_i, \hat{c}_i)$, and the complete set of outputs is denoted as $\hat{\mathbf{y}} = \{\hat{\mathbf{y}}_i\}_{i=1}^N$. Since the number of queries N is larger than K , the proposals are padded to create N tuples. Each bounding box is assigned a label $c_i \in \{0, 1\}$, indicating whether it represents a valid region proposal or a padded entry. The ground truth (GT) is denoted as $\mathbf{y}_i = (\mathbf{b}_i, c_i)$ and the complete set as $\mathbf{y} = \{\mathbf{y}_i\}_{i=1}^K$. To optimally match the proposals with the model predictions, UPicker uses the Hungarian bipartite matching algorithm [26]. In this context, σ denotes the optimal permutation of the prediction indices that minimizes the total matching cost between the GT \mathbf{y} and the predictions $\hat{\mathbf{y}}$. Additionally, CDN is utilized to accelerate model convergence and reduce duplicate detections. The PT loss function is formulated as follows:

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{i=1}^N \left[\lambda_c \mathcal{L}_c(c_i, \hat{c}_{\sigma(i)}) + \lambda_b \mathcal{L}_b(\mathbf{b}_i, \hat{\mathbf{b}}_{\sigma(i)}) \right] + \mathcal{L}_{\text{dn}} \quad (2)$$

where \mathcal{L}_c is the classification loss, implemented via Focal Loss [35], and \mathcal{L}_b combines the \mathcal{L}_1 loss and the Generalized Intersection over Union (GIoU) loss [36]. The term \mathcal{L}_{dn} represents the denoising loss, and the λ_c and λ_b are coefficients that balance the losses.

Fine-tuning stage of UPicker

In the FT stage, UPicker is initialized with pretrained parameters, and then refined using labeled data: $\mathbf{y}_{\text{gt}} = \{(\mathbf{b}_i, c_i)\}_{i=1}^M$, where \mathbf{b}_i represents the bounding box coordinates, and c_i denotes the particle class.

Similarly to the PT stage, the model's prediction heads output both bounding boxes and object categories (particle or background) in the FT stage. UPicker optimizes object matching by identifying the permutation that minimizes the optimal matching cost between the GT particles \mathbf{y}_{gt} and the prediction results $\hat{\mathbf{y}}$. Unlike in the PT stage, the loss is calculated using GT labels in the Hungarian bipartite matching process, and performance metrics are calculated on the validation set for each epoch during FT, leveraging the supervision of GT. The CDN strategy is also employed to enhance model robustness during this stage. To ensure efficient convergence and prevent overfitting, UPicker

utilizes an early stopping mechanism, which terminates training when no further performance improvements are detected after a predefined number of epochs.

Hybrid data augmentation

Since it is difficult to obtain labels for cryo-EM data, UPicker utilizes a hybrid data augmentation strategy during FT, combining offline and online data augmentation techniques to enrich the training set. Offline data augmentation transforms the original training data before feeding it into the model. In contrast, online data augmentation applies transformations randomly during training.

The hybrid data augmentation strategy can increase the size of the training dataset significantly, thus improving the model's accuracy, especially when dealing with limited labeled data (e.g. datasets containing only five labeled images). Cryo-EM images often contain numerous particles with various orientations, and this approach effectively integrates additional information from particles with diverse orientations. Moreover, when online data augmentation alone fails to converge the network training, adding offline augmentation proves to be an effective solution to this limitation.

Contrastive denoising training strategy

The challenge of bipartite matching in cryo-EM particle-picking arises from the high similarity among most particles, often leading to slow convergence and duplicate predictions. The previous methods like TransPicker [18] and CryoTransformer [19] rely on NMS post-processing to eliminate redundant detections. Inspired by DINO [27], UPicker implements CDN, which not only accelerates convergence but also significantly eliminates redundancy, reducing the requirement for extensive post-processing.

As illustrated in Fig. 1d, CDN involves additionally feeding noised GT particles as positive samples into the Transformer decoder and training the model to reconstruct these boxes, meanwhile feeding hard-negative samples and training the model to predict as "no object". During PT, region proposals replace GT particles.

The introduced noise consists of both label noise and bounding box noise. Label noise is injected based on a threshold parameter λ_l . For each GT, if a randomly generated value is less than λ_l , the GT is selected for noise injection, with its label replaced by zero. Bounding box noise is controlled by two thresholds, λ_p and λ_n ($\lambda_p < \lambda_n$), which are employed to generate positive and negative queries. The positive samples are associated with smaller perturbations, while negative samples are with larger ones. To generate challenging negative samples, λ_n is set close to λ_p , thereby enhancing the model's ability to distinguish between similar queries.

The CDN loss comprises two components: $\mathcal{L}_{dn} = \mathcal{L}_p + \mathcal{L}_n$, and

$$\mathcal{L}_p = \lambda_c \mathcal{L}_c(\mathbf{1}, \hat{\mathbf{c}}_p) + \lambda_l \mathcal{L}_l(\mathbf{b}_p, \hat{\mathbf{b}}_p) + \lambda_{\text{GIoU}} \mathcal{L}_{\text{GIoU}}(\mathbf{b}_p, \hat{\mathbf{b}}_p) \quad (3)$$

$$\mathcal{L}_n = \lambda_c \mathcal{L}_c(\mathbf{0}, \hat{\mathbf{c}}_n) \quad (4)$$

where \mathcal{L}_p denotes the reconstruction loss for positive queries, including Focal Loss for classification, \mathcal{L}_l loss, and GIoU loss for bounding box regression. The \mathcal{L}_n represents the classification loss for negative queries, computed using Focal Loss. λ_c , λ_l , and λ_{GIoU} are the control weights. Notably, \mathcal{L}_{dn} is not computed during evaluation.

The two noise types enable UPicker to handle uncertainties of both label and location effectively, thereby enhancing the model's robustness. CDN not only facilitates the model's ability to select

high-quality, nearby queries but also aids in the rejection of more distant ones, thereby mitigating the issues of duplicate predictions and erroneous query selections.

Experiments

Datasets and evaluation metrics

To evaluate the performance of UPicker, experiments were conducted on a diverse set of datasets, including both simulated datasets and experimental datasets. The key characteristics of each dataset are summarized in Table 1.

The experimental datasets include 16 annotated datasets from CryoPPP [38], and the 10470 dataset [39] with particle coordinates downloaded from Electron Microscopy Public Image Archive (EMPIAR) [40]. As summarized in Table 1, these datasets contain between 84 and 300 images and exhibit diversity in protein types, molecular sizes, and particle densities. Figure 2 presents three orthogonal views of the reconstructed 3D models alongside representative micrographs for each dataset, highlighting variations in particle shape. Some datasets are significantly affected by carbon films and contamination. The GT labels may not be entirely accurate, making it challenging to detect all particles, especially in high-noise micrographs. Detailed analysis of these datasets and selection rationale is provided in the Supplementary Materials (see Section S1 and Fig. S1).

To complement the experimental datasets, two simulated datasets were generated using InsilicoTEM software [41], which simulates the cryo-EM imaging process based on physical principles. The simulated datasets utilize the protein structures 3j79.pdb and 1ryp.pdb from the Protein Data Bank (PDB) [42], referred to as SIM-10028(3j79) and SIM-10025(1ryp), respectively. These simulated datasets provide precise particle counts and coordinates.

The particle-picking performance is evaluated using three widely recognized metrics in object detection: precision, recall, and F1-score (i.e. $\frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$). For each dataset, 20% of the data were designated as a separate evaluation set to ensure an unbiased assessment. A selected particle is classified as a true positive if the Intersection over Union (IoU) between its bounding box and the GT bounding box surpasses the threshold of 0.5. Ideally, a good algorithm should have both high precision and high recall, with the F1-score providing a balanced measure of overall performance.

Additionally, we also report the reconstruction resolution of the 3D density maps, as it serves as a critical indicator of the method's effectiveness in accurately identifying particles. Higher resolution values reflect better particle detection and reconstruction quality. Detailed results can be found in Supplementary Section S4.

Comparison with existing methods

UPicker is compared with five particle-picking methods: widely utilized RELION [32], CrYOLO [13], Topaz [14], and two recent deep-learning methods, CryoSegNet [23] and CryoTransformer [19]. While RELION also includes a template matching-based particle-picking approach and an integrated Topaz algorithm, this comparison focused on its LoG-based automatic particle-picking algorithm. For a fair comparison, Topaz, CrYOLO, and UPicker are trained using the same datasets. For CryoTransformer and CryoSegNet, their publicly available trained generic models are adopted. A series of hyperparameter experiments were performed for each method, the best-performing configuration was selected for the final comparison.

Table 1. Datasets used in the experiments. The “#Avg. particles” indicates the average number of labeled particles per image, reflecting the particle density

Dataset	Protein type	#Images	#Avg. particles	Image size(px)	Diameter(px)	Weight(kDa)
SIM-10028	Ribosome (80S)	100	143	4096×4096	160	4000
SIM-10025	T20S proteasome	110	135	3710×3838	140	700
10017	β -galactosidase	84	588	4096×4096	108	450
10075	Bacteriophage MS2	300	42	4096×4096	233	1000
10028	Ribosome (80S)	300	114	4096×4096	224	4000
10077	Ribosome (70S)	300	106	4096×4096	220	2199
10406	Ribosome (70S)	239	93	3838×3710	211	632
10081	Transport protein	300	131	3710×3838	154	298
10590	Transport protein	296	211	3710×3838	158	1000
10093	Membrane protein	295	191	3838×3710	170	779
10096	Viral protein	300	771	3710×3838	84	150
10532	Viral protein	300	293	4096×4096	90	192
10387	Viral protein (DNA)	300	339	3710×3838	213	186
10345	Signaling protein	295	54	3838×3710	149	244
10389	Metal binding protein	300	36	3838×3710	313	1042
10470	Fatty acid synthase	100	32	4096×4096	260	207
10075	Bacteriophage MS2	300	42	4096×4096	233	1000
10947	Viral protein	400	266	4096×4096	240	444
11183	Signaling protein	300	267	5760×4092	159	139

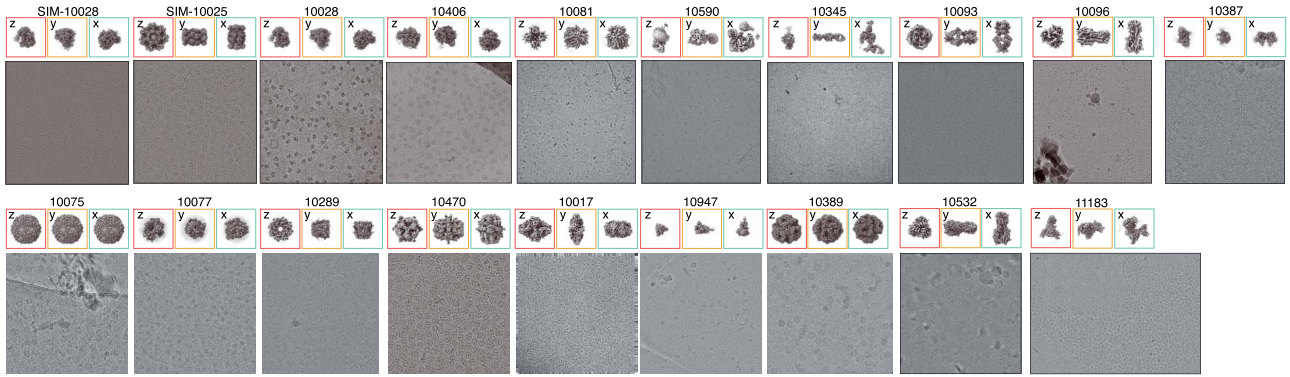


Figure 2. Three orthogonal views of the reconstructed 3D protein particle models for each dataset (obtained from the EMDB [37]), with representative cryo-EM micrographs shown below for each corresponding dataset.

In Fig. 4, a visual comparison of particle-picking results of each software against GT labels is provided. The results indicate that UPicker identified the highest proportion of true particles across these datasets. RELION is sensitive to high-contrast regions, leading to unstable results. Topaz tends to pick more false positives, particularly in contaminated regions, while CrYOLO identifies fewer positive particles. CryoTransformer often produces multiple detections for the same particle and many false positives in contaminated areas. CryoSegNet is difficult to detect smaller particles as well as particles within high-noisy images. Notably, UPicker demonstrates fewer false positives and false negatives compared with other methods, highlighting its high accuracy in detecting target particles. More experimental results are given in Supplementary Section S3 and Figs S3–S5. Interestingly, in cases where experimental data contained inaccurate labels (e.g. dataset 10096), UPicker can identify missing particles and correct erroneous labels.

The statistics of quantitative evaluation are listed in Table 2. Results for several datasets are not included due to concerns regarding the quality of the training labels. UPicker outperforms other methods in terms of Precision, Recall, and F1-score. Even when applied to the challenging EMPIAR-10093 dataset, where

particle shapes are difficult to distinguish, UPicker still achieved notable accuracy. The PR curves for UPicker across some datasets are shown in Fig. 3d. In the box plot shown in Fig. 3e, UPicker consistently achieves higher average Precision and Recall compared with other methods, with a more concentrated distribution and fewer outliers. These results highlight UPicker’s robust and stable performance across diverse datasets.

The computational efficiency of UPicker was compared with other methods on EMPIAR-10081, including training time, inference time, number of training epochs, and GPU memory usage (Table 3). In this evaluation, 100 images were used for training, which is sufficient to achieve high accuracy. CryoSegNet and CryoTransformer were excluded from the comparison due to their reliance on large and diverse datasets for training. The proposed UPicker model was implemented using the PyTorch [43] framework (version 1.12.1). All experiments were executed on a local machine equipped with an Intel Core i7-7820X CPU, 64GB of RAM, and a single NVIDIA GeForce 3090 GPU.

As shown in Table 3, UPicker’s two-stage training process required about 1 hour, which is longer than those of Topaz and CrYOLO. However, UPicker’s FT stage with fewer images converged more quickly while still achieving high accuracy. As

Table 2. Particle-picking results on different evaluation datasets. The Precision, Recall, and F1-score are reported in the corresponding column of each method. Bold font denotes the best average score of each metric

Dataset	Metrics	RELION	Topaz	CrYOLO	CryoTransformer	CryoSegNet	UPicker
SIM-10025	Precision	0.859	0.953	0.978	0.622	0.164	1.000
	Recall	0.695	1.000	1.000	0.779	0.068	1.000
	F1-score	0.769	0.976	0.989	0.692	0.095	1.000
SIM-10028	Precision	0.620	0.990	0.870	0.288	0.728	1.000
	Recall	0.670	0.990	0.750	0.684	0.668	0.990
	F1-score	0.644	0.990	0.805	0.407	0.698	0.995
10028	Precision	0.913	0.765	0.849	0.663	0.781	0.922
	Recall	0.967	0.954	0.946	0.985	0.819	0.965
	F1-score	0.939	0.849	0.895	0.793	0.799	0.943
10406	Precision	0.573	0.710	0.713	0.598	0.800	0.909
	Recall	0.796	0.782	0.926	0.916	0.869	0.998
	F1-score	0.665	0.744	0.805	0.723	0.833	0.951
10081	Precision	0.558	0.684	0.792	0.551	0.662	0.929
	Recall	0.782	0.858	0.788	0.927	0.800	0.982
	F1-score	0.659	0.761	0.790	0.691	0.724	0.955
10096	Precision	0.410	0.599	0.449	0.335	0.664	0.779
	Recall	0.638	0.585	0.588	0.305	0.266	0.850
	F1-score	0.502	0.592	0.509	0.319	0.380	0.813
10345	Precision	0.320	0.435	0.566	0.444	0.509	0.854
	Recall	0.689	0.556	0.513	0.859	0.581	0.983
	F1-score	0.437	0.488	0.538	0.586	0.543	0.912
10590	Precision	0.663	0.525	0.740	0.518	0.749	0.799
	Recall	0.560	0.581	0.695	0.613	0.794	0.868
	F1-score	0.607	0.552	0.717	0.561	0.771	0.832
10532	Precision	0.336	0.391	0.591	0.387	0.391	0.720
	Recall	0.512	0.458	0.591	0.412	0.404	0.852
	F1-score	0.408	0.422	0.591	0.399	0.397	0.781
10093	Precision	0.397	0.367	0.417	0.244	0.273	0.670
	Recall	0.481	0.368	0.372	0.427	0.324	0.676
	F1-score	0.434	0.368	0.393	0.311	0.297	0.673
10075	Precision	0.824	0.892	0.910	0.867	0.887	0.915
	Recall	0.913	0.926	0.945	0.870	0.924	0.980
	F1-score	0.866	0.909	0.927	0.868	0.905	0.946
10077	Precision	0.657	0.785	0.720	0.587	0.698	0.768
	Recall	0.850	0.936	0.879	0.757	0.845	0.994
	F1-score	0.741	0.854	0.792	0.661	0.764	0.867
10289	Precision	0.324	0.586	0.513	0.425	0.386	0.594
	Recall	0.572	0.689	0.723	0.562	0.672	0.769
	F1-score	0.414	0.633	0.600	0.484	0.490	0.670
10017	Precision	0.738	0.848	0.820	0.756	0.813	0.855
	Recall	0.593	0.932	0.602	0.569	0.604	0.950
	F1-score	0.658	0.888	0.694	0.649	0.693	0.900
10947	Precision	0.532	0.635	0.619	0.561	0.528	0.681
	Recall	0.613	0.738	0.662	0.623	0.547	0.771
	F1-score	0.570	0.683	0.640	0.590	0.537	0.723
10389	Precision	0.506	0.868	0.880	0.810	0.880	0.916
	Recall	0.762	0.912	0.930	0.912	0.927	0.999
	F1-score	0.608	0.889	0.904	0.858	0.903	0.956
Average	Precision	0.611	0.694	0.686	0.577	0.659	0.853
	Recall	0.682	0.781	0.735	0.706	0.588	0.927
	F1-score	0.605	0.740	0.759	0.605	0.653	0.895

shown in Fig. 3a and Table 2, UPicker, trained with 50 labeled images, demonstrates superior accuracy compared with other methods trained with 300 images. In terms of inference speed, both UPicker and CrYOLO achieve near real-time performance. Regarding GPU memory usage during training, UPicker, due to its transformer-based architecture, consumes more GPU memory than Topaz but less than CrYOLO, which is based on the TensorFlow framework. To provide a deeper understanding of UPicker’s computational behavior, a detailed analysis of its

time and space complexity is included in the Supplementary Section S2.

Overall, UPicker demonstrates superior particle-picking performance compared to existing methods, combining high detection accuracy with robust generalization across diverse datasets. While UPicker requires a slightly longer training time, its ability to achieve high accuracy with fewer labeled images and its effective handling of challenging datasets underscore its practical applicability and efficiency. Additionally, UPicker is capable of

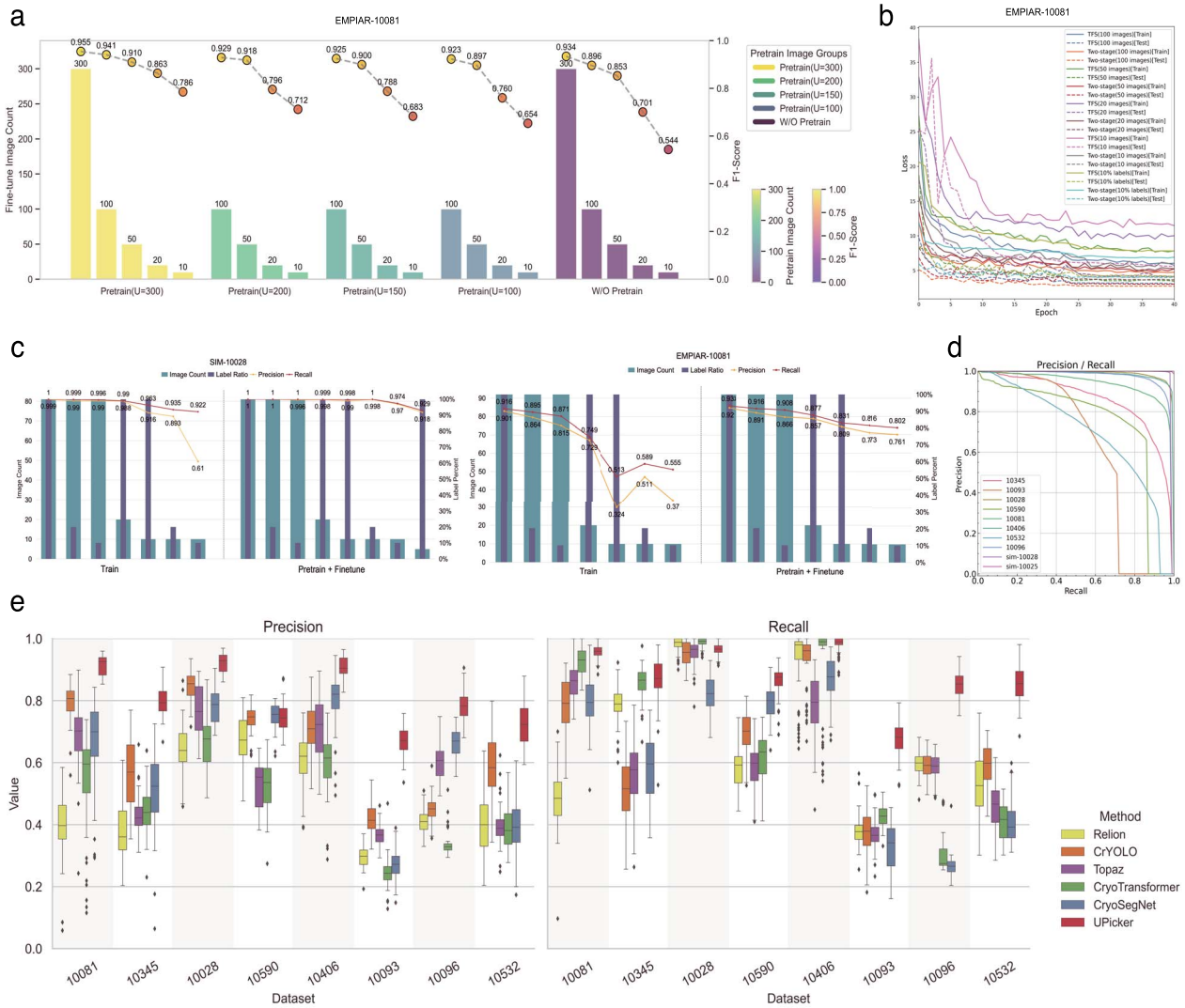


Figure 3. (a) Variation in UPicker's performance on EMPIAR-10081 across different PT and FT image counts. (b) Loss trajectory comparison of UPicker on EMPIAR-10081 across varying training set sizes. (c) Comparison of UPicker's performance under two training modalities across different training set sizes. The bar graphs represent the number of labeled training images and the label ratios, while the line graphs depict the precision and recall results. The left and right sections of the subfigures for SIM-10028 and EMPIAR-10081 illustrate the results of training from scratch and two-stage training, respectively. (d) PR curve comparison of UPicker's performance using all images for two-stage training on ten datasets. (e) Boxplot comparison of precision and recall distributions across six methods on eight datasets.

Table 3. Comparison of computational efficiency on the EMPIAR-10081 dataset (100 images)

Metrics	CrYOLO	Topaz	CryoSegNet	CryoTransformer	UPicker
Batch Size	4	256	6	8	4
Epoch	22	30	200	300	50+50
Training Time	28min	33min	—	—	31+20 min
Training GPU Mem.	17.8G	4.87G	—	—	13.0G
Inference Time	0.31s	2.34s	8.14s	3.13s	0.28s
Inference GPU Mem.	5.32G	4.64G	12.3G	2.16G	3.89G
Model Size	202.7M	1.3M	914.1M	525.8M	561.0M

achieving near real-time inference speed when tested on a single GPU. The implementation also supports multi-GPU setups and the use of larger GPUs with increased batch sizes, allowing for further acceleration of both the training and inference processes when additional hardware resources are available. This flexibility making it a promising tool for high-throughput cryo-EM particle-picking tasks.

Performance evaluation with varying pretraining and fine-tuning data

The performance of UPicker is assessed by varying the number of images used for PT and the amount of labeled data used for FT.

Figure 3a illustrates the impact of PT with different numbers of images from the EMPIAR-10081 dataset, followed by FT on varying numbers of labeled images. The results demonstrate

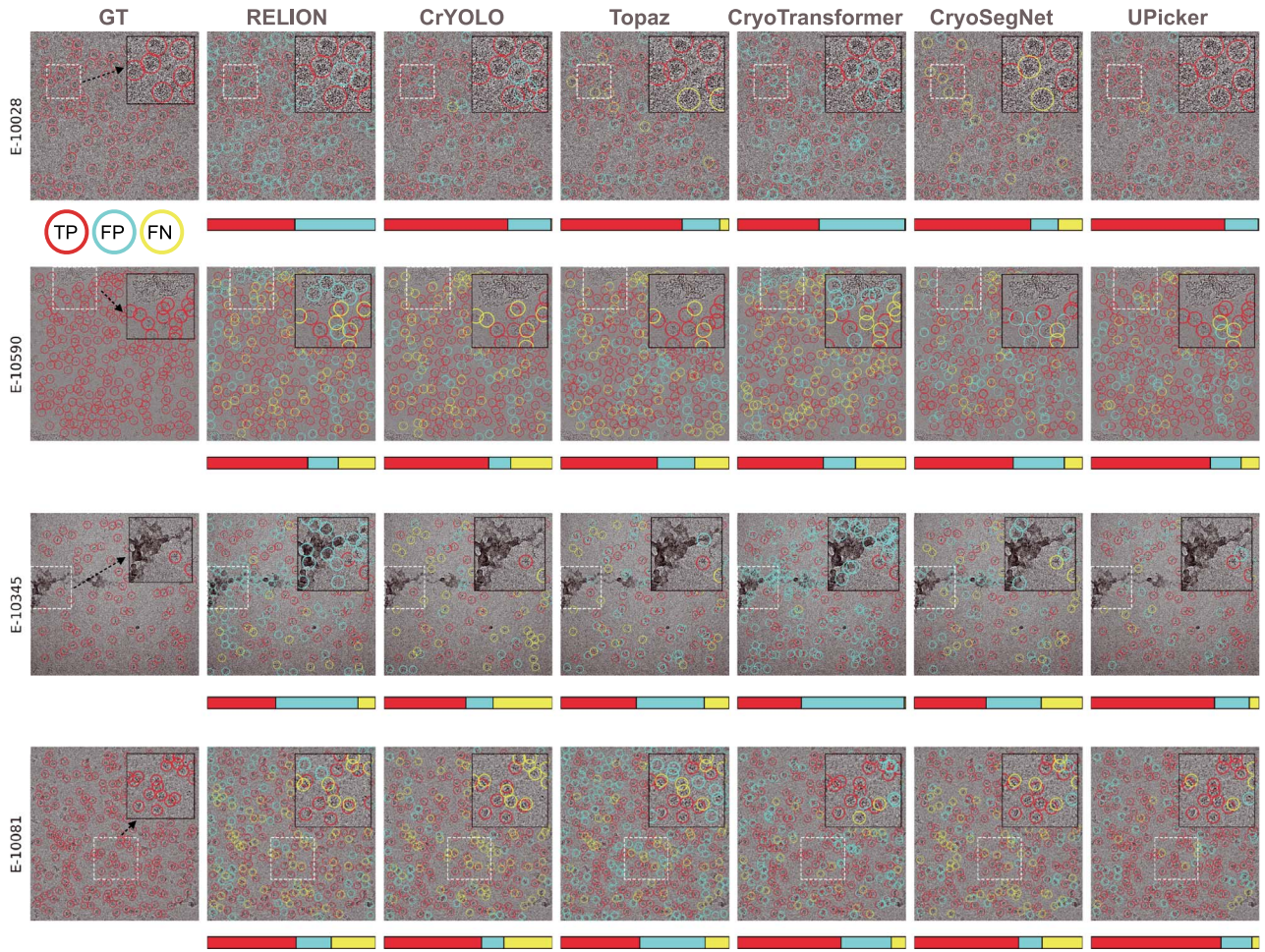


Figure 4. Comparison of particle picking on cryo-EM micrographs from different datasets using six methods (RELION, CrYOLO, Topaz, CryoTransformer, CryoSegNet, and UPicker) with GT labeled particles. The red circles represent the true positive particles (TP) identified by each method, while in the GT column, they represent the actual particle locations. Yellow circles signify missed particles (FN), and cyan circles indicate falsely detected particles (FP). Below each image, a corresponding bar chart illustrates the proportions of TP, FP, and FN, with colors consistent with those of the circles in the figure. The black box area offers an enlarged view of the region outlined by the white dotted box, facilitating detailed comparisons among the various particle picking methods.

that PT generally enhances model performance, especially when labeled data are limited. For larger amounts of labeled training data (e.g. more than 50 images), the model achieves comparable performance even when trained from scratch. Therefore, when labeled data are limited, unsupervised PT significantly improves detection accuracy. Typically, 200 PT images are sufficient to balance training efficiency and accuracy.

To assess how different amounts of labeled data affect performance, two experimental groups were established: two-stage training and training from scratch (TFS). Both groups utilized subsets of SIM-10028 (100 images) and EMPIAR-10081 (114 images) datasets, with variations in image numbers and labeling ratios (Fig. 3c). Identical training configurations were applied to both groups, and the resulting models were evaluated on the same validation datasets. In the two-stage training group, an initial unsupervised PT stage of 50 epochs was performed, followed by FT on the entire set of fully labeled training images. To further assess model robustness, additional experiments were conducted using progressively smaller training datasets, reducing either the number of images or label ratios. Furthermore, combinations of reduced image counts and label ratios were tested to examine model performance under minimal training conditions. In the TFS

group, the model was trained entirely from scratch, with parameters initialized randomly. The labeled data for each dataset were used following the same data configurations as those used in the two-stage training group. This experimental setup allows for a direct and comprehensive comparison of the two approaches and evaluates their effectiveness in scenarios with limited training data.

The results consistently show that performance improves with more training data (Fig. 3c). Notably, the two-stage training approach consistently outperforms TFS, regardless of the amount of labeling. TFS with highly limited labeled data (e.g. 10% labels or 10 images) did not yield successful results, whereas the two-stage approach maintained relatively high precision and recall even with reduced data. For example, on EMPIAR-10081, when trained on only 10 images with a 10% labeled ratio, the two-stage approach achieved a precision of 0.761 and recall of 0.802, compared to 0.370 precision and 0.555 recall for TFS. Figure 3b shows the loss trajectories of UPicker on EMPIAR-10081 for different training set sizes. The two-stage training method demonstrates faster and more significant loss reduction than TFS, with larger training sets yielding faster convergence and lower final loss values.

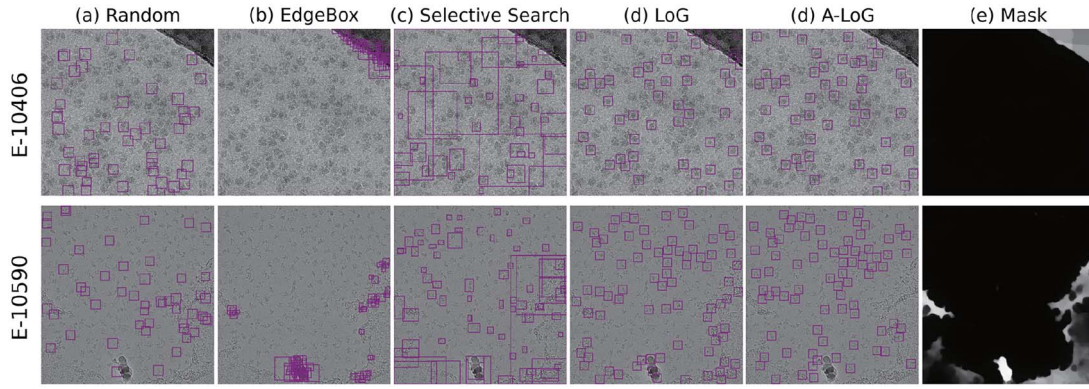


Figure 5. Region proposals obtained by different methods. The last column shows the masks used in the A-LoG region proposal generator.

These experiments highlight the impact of varying data amounts on both PT and FT stages. Overall, these findings underscore the efficiency of UPicker’s two-stage training strategy in maximizing data utilization and enhancing model robustness, especially when labeled data are limited.

Comparison of different region proposal methods

The effectiveness of the proposed A-LoG region proposal method was evaluated against three alternative approaches: selective search [29], EdgeBox [30], and a random region proposal strategy. Moreover, the A-LoG method without the mask filtering step, referred to as the LoG method, was also tested. These methods were applied to pretrain a model on EMPIAR-10028, EMPIAR-10406, and EMPIAR-10389.

For the random region proposal method, 50 square boxes were randomly cropped per image, with sizes corresponding to the estimated range of particle diameters. The selective search method used the “fast” preset from the OpenCV implementation, while the EdgeBox method used the “createEdgeBoxes” function in OpenCV, selecting the top 50 results with the highest confidence scores per image. After PT, the model was fine-tuned using 20 images from each dataset.

Figure 5 visualizes the region proposals generated by each method. The random method generates arbitrary bounding boxes as expected, which, despite lacking precision, still provide shape and size information. The EdgeBox method tends to produce larger candidate boxes that frequently do not correspond to particle positions due to the challenge of identifying particle edges in cryo-EM images, which limits its utility for PT. Similarly, the selective search method is difficult to generate valid candidate regions for particles, as both selective search and EdgeBox rely on features such as color continuity and edge detection that are often absent in cryo-EM images. Consequently, these methods fail to provide precise bounding boxes or effectively localize small objects, which is crucial for particle-picking. In contrast, the LoG region proposal method more effectively generates region proposals corresponding to particle positions, owing to its ability to enhance the SNR and detect particles of varying sizes. Compared with LoG, A-LoG further refines the process by eliminating most proposals in areas affected by carbon film and ice contamination, significantly improving the accuracy of pseudo-labels. The “Mask” in the last column of Fig. 5 refers to the prediction mask employed to filter out unsuitable region proposals.

Quantitative results confirm that the A-LoG region proposal method outperforms the other methods (Table 4). The table also illustrates the impact of each step within the A-LoG method. The

choice of K , the number of top proposals, depends on the dataset being processed. For the 10389 dataset with fewer particles, there are only 36 particles per image on average, and a too-large top- K value actually does not provide additional filtering effects. For images containing a larger number of particles, a larger K value provides more supervision information to the model during PT. However, an excessively large K may introduce false positives and extend PT time. In practice, UPicker sets K to 50 by default.

Ablation studies on UPicker components

To assess the contribution of each component in UPicker, ablation studies were conducted (Table 5). The experiments utilized three benchmark datasets: EMPIAR-10028, EMPIAR-10081, and EMPIAR-10532. Two baselines with distinct data configurations were designed to illustrate UPicker’s effectiveness and flexibility across varying data conditions. UPicker (Baseline 1) utilizes 300 unlabeled images and 100 labeled (L) images, incorporating all components except hybrid data augmentation (AUG). This baseline serves as the reference for optimal performance, representing a scenario with sufficient labeled data to establish a strong benchmark. UPicker (Baseline 2) uses 300 unlabeled and only 20 labeled images, again including all components except hybrid data augmentation. This configuration simulates a low-data environment, allowing us to assess the model’s performance when labeled data are limited and to examine the contribution of each component under such constraints.

Baseline 1 achieved the highest performance across all datasets, with F1-scores of 0.941, 0.987, and 0.950 for EMPIAR-10081, EMPIAR-10028, and EMPIAR-10532, respectively. Despite a significant reduction in labeled data to just 20 images, Baseline 2 still attained an average F1-score of 0.872 across the respective datasets. This demonstrates UPicker’s ability to deliver robust and accurate performance even in scenarios with limited labeled data.

Removing PP caused a significant performance drop, with F1-scores falling to 0.084, 0.033, and 0.135. This shows that PP is crucial for enhancing image quality and preparing the data for subsequent stages.

Omitting the PT stage also resulted in a noticeable performance drop, particularly in the low-data scenario. When using only 20 labeled images without PT, the F1-scores decreased to 0.699, 0.907, and 0.901, compared to 0.763, 0.915, and 0.937 when PT was included. More detailed comparative experiments were carried out in the section on varying data regimes. This highlights the advantage of PT in reducing the need for labeled data.

The absence of FT similarly impacted performance, with F1-scores dropping to 0.421, 0.285, and 0.568, underscoring the

Table 4. The performance of different region proposal methods

Method	Top-K EMPIAR-10028 (U=300)				EMPIAR-10406 (U=239)				EMPIAR-10389 (U=300)				Average			
	Precision	Recall	F1		Precision	Recall	F1		Precision	Recall	F1		Precision	Recall	F1	
W/O PT	-	0.768	0.975	0.859	0.862	0.927	0.893		0.831	0.985	0.901		0.820	0.962	0.884	
Random	50	0.681 (0.087 ↓)	0.882 (0.093 ↓)	0.769 (0.090 ↓)	0.782 (0.080 ↓)	0.949 (0.022 ↑)	0.857 (0.036 ↓)		0.753 (0.078 ↓)	0.993 (0.008 ↑)	0.857 (0.044 ↓)		0.739 (0.081 ↓)	0.941 (0.021 ↓)	0.794 (0.090 ↓)	
Selective	50	0.781 (0.013 ↑)	0.973 (0.002 ↓)	0.866 (0.007 ↑)	0.823 (0.039 ↓)	0.972 (0.045 ↑)	0.891 (0.002 ↑)		0.760 (0.071 ↓)	0.978 (0.007 ↓)	0.855 (0.046 ↓)		0.788 (0.032 ↓)	0.974 (0.012 ↑)	0.871 (0.013 ↓)	
Search																
EdgeBox	50	0.793 (0.025 ↑)	0.988 (0.013 ↑)	0.880 (0.021 ↑)	0.815 (0.047 ↓)	0.982 (0.055 ↑)	0.890 (0.003 ↓)		0.753 (0.078 ↓)	0.970 (0.015 ↓)	0.848 (0.053 ↓)		0.787 (0.033 ↓)	0.980 (0.018 ↑)	0.873 (0.011 ↓)	
LoG	30	0.830 (0.062 ↑)	0.998 (0.023 ↑)	0.906 (0.047 ↑)	0.860 (0.002 ↓)	0.998 (0.071 ↑)	0.924 (0.031 ↑)		0.877 (0.046 ↑)	0.999 (0.014 ↑)	0.934 (0.033 ↑)		0.856 (0.036 ↑)	0.998 (0.036 ↑)	0.921 (0.037 ↑)	
	50	0.832 (0.064 ↑)	0.998 (0.023 ↑)	0.907 (0.048 ↑)	0.869 (0.007 ↑)	0.999 (0.072 ↑)	0.929 (0.036 ↑)		0.875 (0.044 ↑)	0.999 (0.014 ↑)	0.933 (0.032 ↑)		0.859 (0.039 ↑)	0.999 (0.037 ↑)	0.923 (0.039 ↑)	
	80	0.834 (0.066 ↑)	0.999 (0.024 ↑)	0.909 (0.050 ↑)	0.864 (0.002 ↑)	0.999 (0.072 ↑)	0.927 (0.034 ↑)		0.874 (0.043 ↑)	0.999 (0.014 ↑)	0.932 (0.031 ↑)		0.857 (0.037 ↑)	0.999 (0.037 ↑)	0.923 (0.039 ↑)	
All		0.832 (0.064 ↑)	0.997 (0.022 ↑)	0.907 (0.048 ↑)	0.863 (0.001 ↑)	0.999 (0.072 ↑)	0.926 (0.033 ↑)		0.882 (0.051 ↑)	0.999 (0.014 ↑)	0.937 (0.036 ↑)		0.859 (0.039 ↑)	0.998 (0.036 ↑)	0.923 (0.039 ↑)	
A-LoG	30	0.832 (0.064 ↑)	0.997 (0.022 ↑)	0.907 (0.048 ↑)	0.860 (0.002 ↓)	0.998 (0.071 ↑)	0.924 (0.031 ↑)		0.869 (0.038 ↑)	0.998 (0.013 ↑)	0.929 (0.028 ↑)		0.854 (0.034 ↑)	0.998 (0.036 ↑)	0.920 (0.036 ↑)	
	50	0.835 (0.067 ↑)	0.998 (0.023 ↑)	0.909 (0.050 ↑)	0.876 (0.014 ↑)	0.999 (0.072 ↑)	0.933 (0.040 ↑)		0.880 (0.049 ↑)	0.999 (0.014 ↑)	0.936 (0.035 ↑)		0.864 (0.044 ↑)	0.999 (0.037 ↑)	0.926 (0.042 ↑)	
	80	0.838 (0.070 ↑)	0.999 (0.024 ↑)	0.911 (0.052 ↑)	0.874 (0.012 ↑)	0.999 (0.072 ↑)	0.932 (0.040 ↑)		0.882 (0.051 ↑)	0.999 (0.014 ↑)	0.937 (0.036 ↑)		0.865 (0.045 ↑)	0.999 (0.037 ↑)	0.927 (0.043 ↑)	
All		0.832 (0.064 ↑)	0.999 (0.024 ↑)	0.908 (0.049 ↑)	0.874 (0.012 ↑)	0.998 (0.071 ↑)	0.932 (0.039 ↑)		0.883 (0.052 ↑)	0.999 (0.014 ↑)	0.937 (0.036 ↑)		0.863 (0.043 ↑)	0.999 (0.037 ↑)	0.926 (0.042 ↑)	

Table 5. Ablation studies on each component of UPicker with various L values and multiple benchmark datasets

Row	Description	#Img(U)	#Img(L)	PP	PT	FT	CDN	CLS	AUG	EMPIAR-10081 (Prec/Recall/F1)	EMPIAR-10028 (Prec/Recall/F1)	EMPIAR-10389 (Prec/Recall/F1)	Average (Prec/Recall/F1)
1	Baseline 1	300	100	✓	✓	✓	✓	✓	×	0.909/0.975/0.941	0.981/0.993/0.987	0.906/0.999/0.950	0.932/0.989/0.959
2	Baseline 2	300	20	✓	✓	✓	✓	✓	×	0.669/0.889/0.763	0.890/0.942/0.915	0.882/0.999/0.937	0.814/0.943/0.872
3	W/O Preprocess	300	100	×	✓	✓	✓	✓	×	0.062/0.130/0.084	0.018/0.208/0.033	0.102/0.198/0.135	0.061/0.179/0.084
4	W/O Pretrain	0	100	✓	×	✓	✓	✓	×	0.864/0.953/0.905	0.851/0.905/0.877	0.891/0.999/0.942	0.869/0.952/0.908
5	W/O Pretrain	0	20	✓	×	✓	✓	✓	×	0.628/0.793/0.699	0.832/0.998/0.907	0.831/0.985/0.901	0.763/0.925/0.836
6	W/O Fine-tune	300	0	✓	✓	×	✓	✓	×	0.578/0.331/0.421	0.395/0.220/0.285	0.604/0.536/0.568	0.526/0.362/0.425
7	W/O CDN	300	100	✓	✓	✓	✓	✓	×	0.876/0.972/0.922	0.951/0.979/0.964	0.904/0.999/0.949	0.910/0.983/0.945
8	W/O CDN	300	20	✓	✓	✓	×	✓	×	0.652/0.870/0.745	0.811/0.975/0.885	0.875/0.999/0.933	0.779/0.948/0.854
9	W/O \mathcal{L}_c	300	100	✓	✓	✓	✓	×	×	0.020/0.063/0.030	0.079/0.120/0.095	0.068/0.270/0.109	0.056/0.151/0.078
10	W/O \mathcal{L}_c	300	20	✓	✓	✓	✓	×	×	0.046/0.202/0.075	0.144/0.395/0.211	0.078/0.236/0.117	0.089/0.278/0.134
11	W/ HybridAUG	300	20	✓	✓	✓	✓	✓	✓	0.838/0.943/0.887	0.839/0.999/0.912	0.890/0.999/0.941	0.856/0.980/0.913
12	W/ HybridAUG	300	5	✓	✓	✓	✓	✓	✓	0.670/0.902/0.767	0.942/0.977/0.959	0.865/0.996/0.926	0.826/0.977/0.884
13	W/O HybridAUG	300	5	✓	✓	✓	✓	✓	×	0.473/0.765/0.586	0.904/0.960/0.931	0.747/0.990/0.852	0.708/0.905/0.790

importance of this stage for model refinement. This demonstrated that FT is crucial for refining the model to specific datasets.

Additionally, removing the CDN led to a moderate decline in performance, with F1-scores of 0.922, 0.964, and 0.949 for 100 labeled images, and 0.745, 0.885, and 0.933 for 20 labeled images, illustrating its role in enhancing the model's robustness.

Eliminating the classification loss (\mathcal{L}_c) had a severe impact, reducing F1-scores to 0.030, 0.095, and 0.109 with even 100 labeled images, emphasizing its critical contribution to the model's discriminative power. This indicates that the classification loss is vital for the model's discriminative power, allowing it to differentiate between particle and non-particle regions effectively.

Finally, to evaluate the effectiveness of hybrid augmentation strategy (HybridAUG), an extreme low-data scenario with only five labeled images were used. Incorporating HybridAUG significantly improves the F1-scores from 0.586, 0.931, and 0.852 to 0.767, 0.959, and 0.926, underscoring the substantial benefit of advanced augmentation strategy in low-data scenarios.

Overall, these ablation studies confirm that each component of UPicker plays an important role in achieving optimal performance.

Conclusion

This study presents UPicker, a novel cryo-EM particle-picking method that combines an unsupervised PT stage with a subsequent supervised FT stage, utilizing a transformer-based detector. The proposed A-LoG region proposal method effectively generates high-recall pseudo-labels during PT, while the CDN strategy accelerates convergence and reduces redundant detections. Additionally, the hybrid data augmentation strategy further enhances the model's precision and recall in low-data regimes.

Experimental results demonstrate that UPicker achieves superior accuracy compared with state-of-the-art methods, requiring fewer labeled data than transformer-based models and a comparable amount to CNN-based approaches. While its training time is slightly longer than CNN-based methods, it is significantly shorter than other transformer models and offers rapid inference speeds suitable for large-scale applications. UPicker also shows robust performance across diverse datasets, including those with high noise levels and contamination, consistently outperforming existing methods under such challenging conditions. Despite these advancements, UPicker still involves considerable computational demands during training due to its transformer-based architecture. Future research will explore fully unsupervised particle-picking methods that maintain high accuracy. This work provides a new and effective approach for particle-picking in cryo-EM, paving the way for more accurate and efficient structural biology studies.

Key Points

- A semi-supervised transformer-based method (UPicker) is proposed for automatic particle picking in cryo-EM micrographs, utilizing a two-stage training process: unsupervised PT followed by small-scale supervised FT.
- An Adaptive-LoG region proposal generator is designed specifically for cryo-EM micrographs to generate high-recall region proposals, which serve as pseudo-labels during the PT stage.

- UPicker adopts the CDN strategy to accelerate convergence and reduce duplicate detections during the two-stage training. For the FT stage, UPicker utilizes the hybrid data augmentation strategy to improve accuracy with limited labeled data.
- UPicker outperforms state-of-the-art particle-picking methods in both accuracy and robustness, achieving high detection performance even with limited labeled data, thereby facilitating the analysis of high-resolution structures in cryo-EM.

Supplementary data

Supplementary data are available at *Briefings in Bioinformatics* online.

Funding

This research was supported by the National Natural Science Foundation of China under Grant [61932018], and the National Key Research and Development Program of China [2021YFF0704300].

Data availability

The public datasets used in this study were obtained from <https://github.com/BioinfoMachineLearning/cryoppp> and <https://www.ebi.ac.uk/empir>. The source code is available at <https://github.com/JachyLikeCoding/UPicker>.

Conflict of interest

None declared.

References

1. Vilas JL, Carazo JM, Carlos Oscar S. et al. Emerging themes in cryoem—single particle analysis image processing. *Chem Rev* 2022;**122**:13915–51. <https://doi.org/10.1021/acs.chemrev.1c00850>.
2. Bendory T, Bartesaghi A, Singer A. Single-particle cryo-electron microscopy: mathematical theory, computational challenges, and opportunities. *IEEE Signal Process Mag* 2020;**37**:58–76. <https://doi.org/10.1109/MSP.2019.2957822>.
3. Moriya T, Saur M, Stabrin M. et al. High-resolution single particle analysis from electron cryo-microscopy images using SPHIRE. *J Vis Exp* 2017;**123**:e55448. <https://doi.org/10.3791/55448>.
4. Sorzano COS, Marabini R, Velázquez-Muriel J. et al. Xmipp: a new generation of an open-source image processing package for electron microscopy. *J Struct Biol* 2004;**148**:194–204. <https://doi.org/10.1016/j.jsb.2004.06.006>.
5. Huang Z, Penczek PA. Application of template matching technique to particle detection in electron micrographs. *J Struct Biol* 2004;**145**:29–40. <https://doi.org/10.1016/j.jsb.2003.11.004>.
6. Roseman AM. FindEM—a fast, efficient program for automatic selection of particles from electron micrographs. *J Struct Biol* 2004;**145**:91–9. <https://doi.org/10.1016/j.jsb.2003.11.007>.
7. Tang G, Peng L, Baldwin PR. et al. EMAN2: an extensible image processing suite for electron microscopy. *J Struct Biol* 2007;**157**:38–46. <https://doi.org/10.1016/j.jsb.2006.05.009>.

8. Lander GC, Stagg SM, Voss NR. et al. Appion: an integrated, database-driven pipeline to facilitate em image processing. *J Struct Biol* 2009;**166**:95–102. <https://doi.org/10.1016/j.jsb.2009.01.002>.
9. Scheres SHW. Relion: implementation of a Bayesian approach to cryo-EM structure determination. *J Struct Biol* 2012;**180**:519–30. <https://doi.org/10.1016/j.jsb.2012.09.006>.
10. Wang F, Gong H, Liu G. et al. DeepPicker: a deep learning approach for fully automated particle picking in cryo-EM. *J Struct Biol* 2016;**195**:325–36. <https://doi.org/10.1016/j.jsb.2016.07.006>.
11. Xiao Y, Yang G. A fast method for particle picking in cryo-electron micrographs based on fast R-CNN. In: *AIP Conference Proceedings*. Melville, NY: AIP Publishing LLC. Vol. **1836**, 2017, p. 020080.
12. Al-Azzawi A, Ouadou A, Tanner JJ. et al. Autocryopicker: an unsupervised learning approach for fully automated single particle picking in cryo-EM images. *BMC Bioinformatics* 2019;**20**:1–26.
13. Wagner T, Merino F, Stabrin M. et al. SPHIRE-crYOLO is a fast and accurate fully automated particle picker for cryo-EM. *Commun Biol* 2019;**2**:218. <https://doi.org/10.1038/s42003-019-0437-z>.
14. Bepler T, Morin A, Rapp M. et al. Positive-unlabeled convolutional neural networks for particle picking in cryo-electron micrographs. *Nat Methods* 2019;**16**:1153–60. <https://doi.org/10.1038/s41592-019-0575-8>.
15. Tegunov D, Cramer P. Real-time cryo-electron microscopy data preprocessing with warp. *Nat Methods* 2019;**16**:1146–52. <https://doi.org/10.1038/s41592-019-0580-y>.
16. Ouyang J, Wang J, Wang Y. et al. CenterPicker: an automated cryo-EM single-particle picking method based on center point detection. *J Cybersecur* 2022;**2**:65.
17. Carion N, Massa F, Synnaeve G. et al. End-to-end object detection with transformers. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Cham, Switzerland: Springer, 2020, 213–29.
18. Zhang C, Li H, Wan X. et al. TransPicker: a transformer-based framework for particle picking in cryoEM micrographs. In: *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. New York, NY: IEEE, 2021, 1179–84.
19. Dhakal A, Gyawali R, Wang L. et al. Cryotransformer: a transformer model for picking protein particles from cryo-EM micrographs. *Bioinformatics* 2024;**40**:btac109.
20. Han X, Zhang Z, Ding N. et al. Pre-trained models: past, present and future. *AI Open* 2021;**2**:225–50. <https://doi.org/10.1016/j.aiopen.2021.08.002>.
21. Dang T, Kornblith S, Nguyen HT. et al. A study on self-supervised object detection pretraining. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 86–99. Cham, Switzerland: Springer, 2022.
22. Zhang X, Zhao T, Chen J. et al. EPicker is an exemplar-based continual learning approach for knowledge accumulation in cryoEM particle picking. *Nat Commun* 2022;**13**:1–10. <https://doi.org/10.1038/s41467-022-29994-y>.
23. Gyawali R, Dhakal A, Wang L. et al. CryoSegNet: accurate cryo-EM protein particle picking by integrating the foundational AI image segmentation model and attention-gated U-Net. *Brief Bioinform* 2024;**25**. <https://doi.org/10.1093/bib/bbae282>.
24. He K, Zhang X, Ren S. et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. New York, NY: IEEE, 2016, 770–8.
25. Zhu X, Weijie S, Lu L. et al. Deformable DETR: deformable transformers for end-to-end object detection. In: *Proceedings of the International Conference on Learning Representations (ICLR)*, pp. 1–16, 2020.
26. Kuhn HW. The Hungarian method for the assignment problem. *Nav Res Logist Q* 1955;**2**:83–97. <https://doi.org/10.1002/nav.3800020109>.
27. Zhang H, Li F, Liu S. et al. DINO: DETR with improved denoising anchor boxes for end-to-end object detection. In: *Proceedings of the International Conference on Learning Representations*, 2022.
28. Alexe B, Deselaers T, Ferrari V. What is an object? In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, New York, NY: IEEE, 2010, 73–80.
29. Uijlings JRR, Van De Sande KEA, Gevers T. et al. Selective search for object recognition. *Int J Comput Vis* 2013;**104**:154–71. <https://doi.org/10.1007/s11263-013-0620-5>.
30. Zitnick CL, Dollár P. Edge boxes: locating object proposals from edges. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Cham, Switzerland: Springer, 2014, 391–405.
31. Ren S, He K, Girshick R. et al. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans Pattern Anal Mach Intell* 2016;**39**:1137–49. <https://doi.org/10.1109/TPAMI.2016.2577031>.
32. Zivanov J, Nakane T, Forsberg BO. et al. New tools for automated high-resolution cryo-EM structure determination in RELION-3. *Elife* 2018;**7**:e42166. <https://doi.org/10.7554/eLife.42166>.
33. Punjani A, Rubinstein JL, Fleet DJ. et al. CryoSPARC: algorithms for rapid unsupervised cryo-EM structure determination. *Nat Methods* 2017;**14**:290–6. <https://doi.org/10.1038/nmeth.4169>.
34. Sanchez-Garcia R, Segura J, Maluenda D. et al. Micrograph-Cleaner: a Python package for cryo-EM micrograph cleaning using deep learning. *J Struct Biol* 2020;**210**:107498. <https://doi.org/10.1016/j.jsb.2020.107498>.
35. Lin T-Y, Goyal P, Girshick R. et al. Focal loss for dense object detection. In: *Proceedings of IEEE International Conference on Computer Vision (ICCV)*. New York, NY: IEEE, 2017, 2980–8.
36. Rezatofighi H, Tsoi N, Gwak JY. et al. Generalized intersection over union: a metric and a loss for bounding box regression. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. New York, NY: IEEE, 2019, 658–66.
37. The wwPDB Consortium. EMDB—the electron microscopy data bank. *Nucleic Acids Res* 2024;**52**:D456–65. <https://doi.org/10.1093/nar/gkad1019>.
38. Dhakal A, Gyawali R, Wang L. et al. A large expert-curated cryo-EM image dataset for machine learning protein particle picking. *Sci Data* 2023;**10**:392. <https://doi.org/10.1038/s41597-023-02280-2>.
39. Singh K, Graf B, Linden A. et al. Discovery of a regulatory subunit of the yeast fatty acid synthase. *Cell* 2020;**180**:1130–1143.e20. <https://doi.org/10.1016/j.cell.2020.02.034>.
40. Iudin A, Korir PK, Salavert-Torres J. et al. EMPIAR: a public archive for raw electron microscopy image data. *Nat Methods* 2016;**13**:387–8. <https://doi.org/10.1038/nmeth.3806>.
41. Vulović M, Ravelli RBG, van Vliet LJ. et al. Image formation modeling in cryo-electron microscopy. *J Struct Biol* 2013;**183**:19–32. <https://doi.org/10.1016/j.jsb.2013.05.008>.
42. Burley SK, Berman HM, Kleywegt GJ. et al. Protein data bank (PDB): the single global macromolecular structure archive. *Protein Crystallography: Methods and Protocols*. Cham: Springer, 2017, 627–41. https://doi.org/10.1007/978-1-4939-7000-1_26.
43. Paszke A, Gross S, Massa F. et al. PyTorch: an imperative style, high-performance deep learning library. In: *Advances in Neural Information Processing Systems 32*, Red Hook, NY: Curran Associates, Inc. 2019, 8024–35.