At FullTilt: Real-Time Open-Set 3D Macromolecule Detection Directly from Tilted 2D Projections

Ming-Yang Ho
Duke University
ming-yang.ho@duke.edu

Alberto Bartesaghi
Duke University
alberto.bartesaghi@duke.edu

Abstract

Open-set 3D macromolecule detection in cryogenic electron tomography eliminates the need for target-specific model retraining. However, strict VRAM constraints prohibit processing an entire 3D tomogram, forcing current methods to rely on slow sliding-window inference over extracted subvolumes. To overcome this, we propose FullTilt, an end-to-end framework that redefines 3D detection by operating directly on aligned 2D tilt-series. Because a tilt-series contains significantly fewer images than slices in a reconstructed tomogram, FullTilt eliminates redundant volumetric computation, accelerating inference by orders of magnitude. To process the entire tilt-series simultaneously, we introduce a tilt-series encoder to efficiently fuse cross-view information. We further propose a multi-class visual prompt encoder for flexible prompting, a tilt-aware query initializer to effectively anchor 3D queries, and an auxiliary geometric primitives module to enhance the model's understanding of multi-view geometry while improving robustness to adverse imaging artifacts. Extensive evaluations on three real-world datasets demonstrate that FullTilt achieves state-of-the-art zero-shot performance while drastically reducing runtime and VRAM requirements, paving the way for rapid, large-scale visual proteomics analysis. Code and data are available at https://github.com/Kaminyou/FullTilt.

1 Introduction

Determining the 3D structures of macromolecules (hereafter referred to as *particles*) is critical for understanding biological mechanisms and advancing drug development [1, 2]. Cryogenic electron tomography (cryo-ET) is an imaging technique that enables *in situ* 3D structural visualization [3, 4]. Using this method, cellular samples are rotated and imaged by capturing a series of 2D projections at different angles. The resulting *tilt-series* is aligned to a common vertical tilt axis ($x = 0.5$ in normalized image coordinates) and subsequently reconstructed into a 3D tomogram. Off-the-shelf computational methods are applied to detect target particles [5], followed by subtomogram averaging and refinement to produce high-resolution structures [4].

While conventional closed-set detection methods [6–8] require laborious data annotation and retraining for every new target, recent work has explored open-set detection paradigms [9–12]. By supplying a *visual prompt* (e.g., a reference subvolume), these frameworks localize arbitrary targets by comparing prompt embeddings against 3D subvolumes extracted from the tomogram. This enables *zero-shot* detection without target-specific retraining.

However, strict VRAM constraints prohibit loading an entire 3D tomogram into memory. As a result, current methods rely on sliding-window inference approaches [9, 11], resulting in long inference runtimes (Figs. 1 and 2a). Others apply SAM [13] to 2D slices along each orthogonal axis of the 3D tomogram [10], but they still exhibit high computational complexity that scales with the tomogram's volumetric dimensions (Figs. 1 and 2b).

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Considering that a 3D tomogram is reconstructed from the aligned 2D tilt-series [14, 15], the tilt-series intrinsically contains at least as much information as the tomogram. Moreover, the number of images in a tilt-series (e.g., 41) is usually smaller than the depth dimension of the reconstructed tomogram (e.g., 256). This indicates that tomograms contain redundant information, which wastes computational resources and slows down processing, raising the critical question: *Can we conduct open-set 3D macromolecule detection directly on the aligned 2D tilt-series to expedite runtime?*

To answer this question, we propose *FullTilt*, an open-set 3D macromolecule detection framework that processes the 2D tilt-series in an end-to-end manner (Fig. 2d). By dramatically reducing the input dimensionality, our framework operates *at full tilt*—running hundreds to thousands of times faster than current state-of-the-art methods (Fig. 1).

FullTilt adopts a DETR-like architecture [16]. We first utilize a 2D backbone to extract initial 2D features [17]. Our tilt-series encoder ($\mathcal{E}_{\text{Tilt}}$) then merges these feature maps into 3D-aware features with an efficient global row attention, followed by a multiclass visual prompt encoder ($\mathcal{E}_{\text{Prompt}}$) that extracts prototypes for the user's requested classes. We then effectively initialize 3D anchors with a tilt-aware query initializer ($\mathcal{I}_{\text{Tilt}}$) and leverage a 3D decoder to obtain the 3D coordinates and labels of the targeted particles. To enhance the model's understanding of multi-view geometry and improve its robustness to adverse imaging artifacts such as occlusion and non-uniform illumination (Fig. 3), we introduce an auxiliary geometric primitives module ($\mathcal{G}$) that generates geometric data *on-the-fly*.

Although 3D macromolecule detection from 2D tilt-series shares similarities with multi-view 3D object detection in autonomous driving [18], directly applying existing frameworks (e.g., DETR3D [19] or PETR [20]) yields poor results, rendering such adaptations non-trivial. This is due to the extremely low signal-to-noise ratio (SNR) characteristic of cryo-ET data [21]. Unlike natural images where objects are clearly visible in a single frame, macromolecules often only become distinguishable when observing consecutive frames across the tilt-series (Fig. 3). For the same reason, naive implementations, such as applying a 2D detector to each tilt image independently and back-projecting the results into 3D space [22], also fail (Figs. 1 and 2c). FullTilt successfully addresses this challenge by introducing a dedicated tilt-series encoder, enabling the model to holistically contextualize all tilts at once.

Evaluated on three public, real-world cryo-ET datasets as well as simulated datasets, FullTilt achieves comparable or superior zero-shot performance against existing open-set baselines, while dramatically reducing runtime and VRAM requirements. In summary, our contributions are as follows:

- To the best of our knowledge, this is the first study to conduct 3D macromolecule detection directly from 2D tilt-series in an end-to-end manner, achieving state-of-the-art zero-shot performance on real-world cryo-ET data while drastically reducing computational overhead.
- We introduce a tilt-series encoder to efficiently merge information across images, paired with a tilt-aware query initializer to effectively establish 3D anchors. Additionally, an auxiliary geometric primitives module is introduced to enhance the model's understanding of multi-view geometry and improve its robustness to adverse imaging artifacts.
- We propose a multiclass visual prompt encoder designed to flexibly handle diverse prompting scenarios (accommodating varying prompt counts and multiple classes per tilt image), which can be seamlessly integrated into any DETR-like framework.

2 Related Work

3D Macromolecule Detection for Cryo-ET. Deep learning has largely superseded template matching in cryo-ET [7]. Methods like crYOLO [6] and PickYOLO [23] merge 2D slice detections into 3D coordinates. Zeng's method [22] back-projects 2D tilt-series detections into 3D, but suffers from high projection errors and lacks cross-tilt fusion. Volumetric frameworks (e.g., DeepFinder [7], DeePiCt [24], VP-Detector [25]) utilize 3D CNNs, requiring sliding-window inference and segmentation clustering. To reduce annotation costs, weakly supervised learning (WSL) methods (Huang's method [26], TomoPicker [12]) employ positive-unlabeled learning [27]. Crucially, all these approaches operate in a *closed-set* paradigm, requiring target-specific model retraining for novel particle classes.

Open-Set 3D Macromolecule Detection. To bypass retraining, open-set detection paradigms have recently emerged [9, 11, 10]. TomoTwin [9] compares a target subvolume's embedding against all

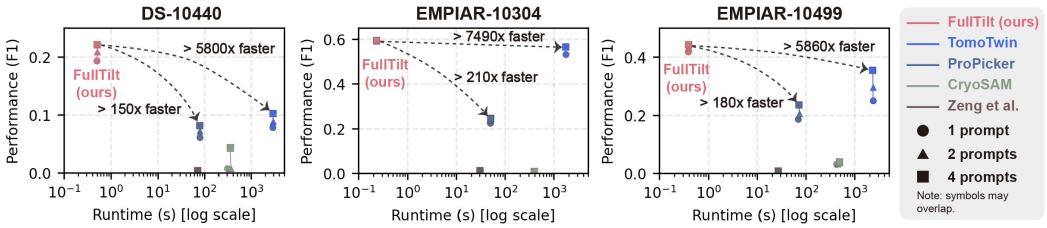


Figure 1: **Runtime and performance comparison.** FullTilt achieves inference speeds hundreds to thousands of times faster than current state-of-the-art open-set macromolecule detection methods, while delivering comparable or superior zero-shot performance across three real-world cryo-ET datasets. The exact acceleration multiplier varies based on the tilt-series and tomogram dimensions.

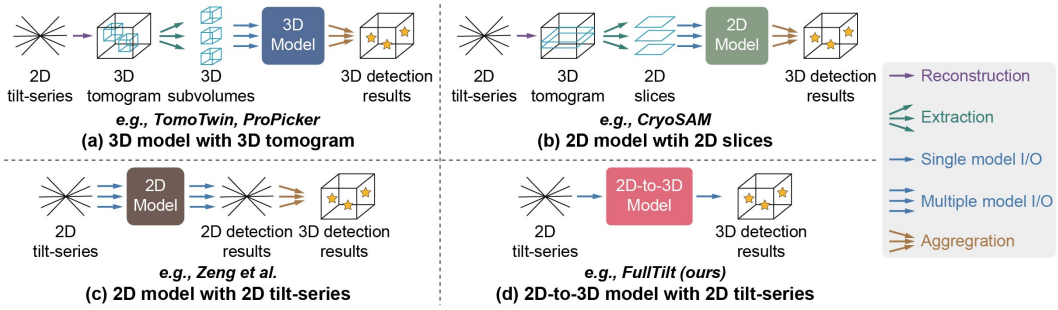


Figure 2: **Comparison of detection mechanisms.** We show existing architectures and inputs for open-set 3D macromolecule detection in cryo-ET (prompting modules omitted for simplicity). (a) TomoTwin [9] and ProPicker [11] process 3D tomograms using 3D models. Due to strict VRAM limits, they rely on sliding-window subvolume extraction, which requires extensive model I/O and bottlenecks inference speed. (b) CryoSAM [10] applies a 2D SAM model to orthogonal 2D slices of the 3D tomogram, necessitating repeated, costly model I/O operations. (c) Zeng et al. [22] apply a 2D detector to individual 2D tilt images and back-project the results into 3D space. This approach suffers from low single-image visibility and severe back-projection errors. (d) FullTilt is the first framework to process the 2D tilt-series directly into 3D detection results in an end-to-end manner.

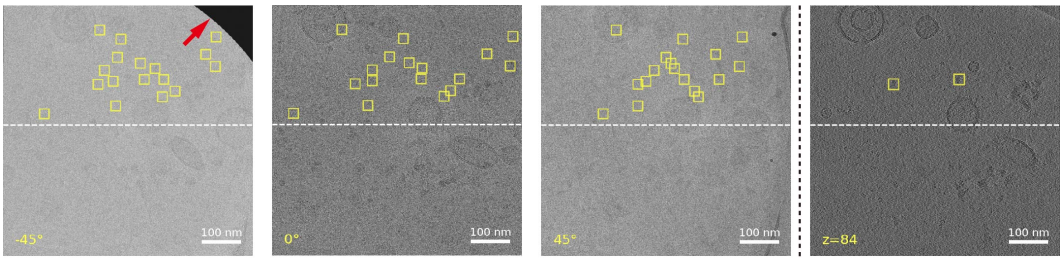
3D tomogram subvolumes. ProPicker [11] frames this as conditional segmentation to improve speed, while CryoSAM [10] clusters masks generated by applying 2D SAM [13] to orthogonal tomogram slices. However, all these methods rely on exhaustive 2D or 3D tomogram patching, which severely bottlenecks inference via repetitive model I/O. Instead, FullTilt directly processes the 2D tilt-series end-to-end, completely bypassing 3D tomogram inference, thus drastically accelerating detection.

Multi-View 3D Object Detection. End-to-end multi-view 3D object detection is prominent in autonomous driving [28]. DETR3D [19] extends DETR [16] for multi-view data, while PETR [20] replaces DETR3D's 3D-to-2D projection with a 3D positional encoder. Subsequent models like PETRv2 [29] incorporate temporal modeling. FullTilt combines DETR3D's training efficiency (via 3D-to-2D projection) with PETR's idea of 3D-aware positional encoding (detailed in Appendix A). Crucially, to handle high-noise cryo-ET data and enable open-set detection, FullTilt further introduces four novel components: a tilt-series encoder ($\mathcal{E}_{\text{Tilt}}$), a multiclass visual prompt encoder ($\mathcal{E}_{\text{Prompt}}$), a tilt-aware query initializer ($\mathcal{I}_{\text{Tilt}}$), and an auxiliary geometric primitives module ($\mathcal{G}$).

3 FullTilt

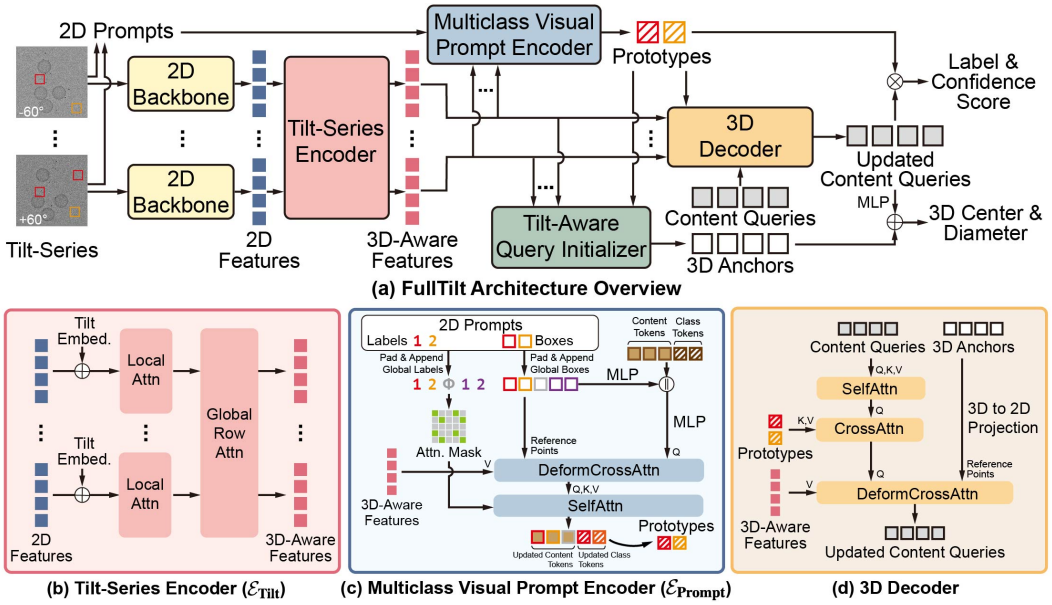
3.1 Problem Definition

Given an aligned 2D tilt-series of N images $\{\mathbf{X}_i\}_{i=1}^{N}$ with $\mathbf{X}_i \in \mathbb{R}^{H\times W}$ and corresponding tilt angles $\{\theta_i\}_{i=1}^{N}$, each image $\mathbf{X}_i$ is accompanied by K_i visual prompts $\{(\mathbf{P}_i^k, Y_i^k)\}_{k=1}^{K_i}$. Each prompt comprises a class label $Y_i^k \in \mathbb{Z}^+$ and a bounding box $\mathbf{P}_i^k = (x, y, d, d) \in \mathbb{R}^4$ defining the particle's 2D center coordinates (x, y) and diameter d.



(a) Images from a tilt-series (b) Tomogram (one slice)

Figure 3: **2D tilt-series vs. 3D tomogram.** We show images at varying tilt angles alongside a tomogram slice reconstructed from the same tilt-series (DS-10440 dataset, `TS_6_6`). Yellow boxes denote thyroglobulin particles (top half). (a) High noise makes recognizing individual particles in a single 2D tilt image difficult without viewing consecutive frames. Furthermore, particles often overlap at certain angles because each image projects through the full sample depth, and raw images often exhibit severe physical occlusion (red arrow). (b) Conversely, within a tomogram Z-slice, particles are separated along the depth axis with no spatial overlap.



(a) FullTilt Architecture Overview

(b) Tilt-Series Encoder ($\mathcal{E}_{\text{Tilt}}$) (c) Multiclass Visual Prompt Encoder ($\mathcal{E}_{\text{Prompt}}$) (d) 3D Decoder

Figure 4: **FullTilt architecture.** (a) Overview of the FullTilt architecture. (b) The tilt-series encoder ($\mathcal{E}_{\text{Tilt}}$) utilizes alternating local and global row attention to fuse information from the 2D features into 3D-aware features. (c) The multiclass visual prompt encoder ($\mathcal{E}_{\text{Prompt}}$) accommodates flexible prompting scenarios by leveraging masked attention to generate prototypes for all requested classes. (d) The 3D decoder updates the content queries via self-attention, cross-attention with the prototypes, and deformable cross-attention with the 3D-aware features, using 2D coordinates projected from the 3D anchors.

FullTilt processes these inputs end-to-end to simultaneously estimate a set of M 3D particles $\{(\hat{\mathbf{P}}^m, \hat{Y}^m, s^m)\}_{m=1}^{M}$. Here, $\hat{\mathbf{P}}^m = (\hat{x}, \hat{y}, \hat{z}, \hat{d}) \in \mathbb{R}^4$ denotes the estimated 3D center and diameter, $\hat{Y}^m \in \mathbb{Z}^+$ is the estimated class label, and $s^m \in [0, 1]$ is the confidence score.

3.2 Architecture Overview

FullTilt follows the DETR design principles [16], using a transformer encoder-decoder to estimate a set of objects from learnable queries, with bipartite matching for training. It consists of five sequential modules (Fig. 4a). First, a **2D backbone** independently extracts multi-level 2D feature

maps from each tilt image. A **tilt-series encoder ($\mathcal{E}_{\text{Tilt}}$)** then collectively processes these features, merging cross-tilt information into 3D-aware features. Subsequently, a **multiclass visual prompt encoder ($\mathcal{E}_{\text{Prompt}}$)** extracts prototypes for the requested classes based on the 2D prompts. Using these prototypes and the 3D-aware features, a **tilt-aware query initializer ($\mathcal{I}_{\text{Tilt}}$)** establishes 3D anchors and initializes learnable content queries. Finally, a **3D decoder** refines these queries to estimate target 3D coordinates, diameters, and labels.

3.3 Architecture Details

2D Backbone. Following the standard DETR design [30], a 2D backbone (e.g., Swin Transformer) extracts multi-level feature maps from each 2D tilt image $\mathbf{X}_i$, denoted as $\mathbf{F}_{i,l} \in \mathbb{R}^{H_l \times W_l \times C}$, where l indicates the feature level and C is the hidden channel dimension used throughout the architecture.

Tilt-Series Encoder ($\mathcal{E}_{\text{Tilt}}$). To lift these 2D features into 3D-aware representations, we aggregate information across all tilt images (Fig. 4b). For a given tilt angle θ, we generate a tilt embedding $\mathbf{e}_\theta \in \mathbb{R}^C$ via $\mathbf{e}_\theta = \text{MLP}\Big(\big[\sin(\theta\omega_0), \ldots, \sin(\theta\omega_{C/2-1}), \cos(\theta\omega_0), \ldots, \cos(\theta\omega_{C/2-1})\big]\Big)$, where $\omega_k = 2^{\frac{8k}{C/2-1}}$. These tilt embeddings, alongside 2D positional and level embeddings, are added to the 2D features. We then adapt an alternating-attention (AA) strategy [31], which alternates local (single-image) and global (cross-image) attention to fuse information across images. Specifically, we perform *local attention* within each tilt image's features followed by *global row attention* across all tilts. Since the tilt-series is aligned to a common vertical tilt axis, a target particle consistently occupies the same row across all images. Exploiting this geometric constraint drastically reduces the computational overhead of standard AA—vital given the large number of views N in cryo-ET—while remaining straightforward to implement (Appendix Alg. A1). After L_{AA} layers, we obtain the final 3D-aware features $\{\mathbf{Z}_i\}_{i=1}^N$, where $\mathbf{Z}_i \in \mathbb{R}^{\tilde{H} \times \tilde{W} \times C}$ (multi-level notation omitted for simplicity).

Multiclass Visual Prompt Encoder ($\mathcal{E}_{\text{Prompt}}$). Given the 2D prompts $\{\{(\mathbf{P}_i^k, Y_i^k)\}_{k=1}^{K_i}\}_{i=1}^N$, let $N_{\text{p}} := \max_i K_i$ be the maximum number of prompts across all images and N_{c} be the total number of unique classes. For each $i \in \{1, \ldots, N\}$, we pad the local prompt labels $\{Y_i^k\}_{k=1}^{K_i}$ to size N_{p} using an empty token ϕ, and append the N_{c} unique global labels $(1, \ldots, N_c)$ to form the label vector $\mathbf{V}_i$ of size $N_{\text{p}} + N_{\text{c}}$ (while we assume ordered labels here for simplicity, our framework supports arbitrary $\mathbb{Z}^+$ labels; details in Appendix B.1). To govern information flow, we generate an attention mask $\mathbf{A}_i \in \{0,1\}^{(N_{\text{p}}+N_{\text{c}}) \times (N_{\text{p}}+N_{\text{c}})}$ (Fig. 4c), where the entry at row r and column c permits attention only between valid, identical labels (i.e., $\mathbf{V}_i[r] = \mathbf{V}_i[c] \neq \phi$).

Similarly, we pad the local bounding boxes $\{\mathbf{P}_i^k\}_{k=1}^{K_i}$ to size N_{p} using dummy coordinates $(0, 0, 0, 0)$. Extending the global bounding box approach introduced in T-Rex2 [32], we append N_{c} such boxes $(0.5, 0.5, 1.0, 1.0)$ and encode the entire spatial set via an MLP. Next, we initialize N_{p} learnable *content tokens* and N_{c} learnable *class tokens*. These are concatenated along the channel dimension with their corresponding encoded boxes and fused via an MLP to form a unified token sequence $\mathbf{E}_i \in \mathbb{R}^{(N_{\text{p}}+N_{\text{c}}) \times C}$. This sequence is iteratively refined through L_{VP} layers. Each layer applies deformable cross-attention with the 3D-aware features, followed by masked self-attention utilizing $\mathbf{A}_i$, which enables the class tokens to aggregate information exclusively from relevant content tokens. After each iteration, the updated class tokens are averaged across all tilt images containing that specific class. The final extracted class tokens serve as our class *prototypes*, denoted as $\mathbf{t} \in \mathbb{R}^{N_{\text{c}} \times C}$.

Tilt-Aware Query Initializer ($\mathcal{I}_{\text{Tilt}}$). Because a particle's x and y coordinates align perfectly with its projection at a 0° tilt, we strategically initialize the *3D anchors* $\{\mathbf{q}^m_{\text{Anchor}}\}_{m=1}^M$ using the near-zero tilt image. We first extract the features $\mathbf{Z}_{i^*}$ from the tilt closest to 0°, where $i^* = \arg\min_i |\theta_i|$. Following previous work [32, 33], we select the initial 2D anchors (x, y, d, d) from the bounding boxes corresponding to the top M entries of $\mathbf{S} \in \mathbb{R}^{\tilde{H} \times \tilde{W}}$, where $S_{h,w} = \max_{1 \leq j \leq N_c} (\mathbf{Z}_{i^*} \mathbf{t}^\top)_{h,w,j}$ takes the maximum over the N_c class prototypes at each spatial location. Assuming a zero-mean depth bias within the tomogram, we initialize the z-coordinate to 0.5 (the middle plane), yielding 3D anchors formatted as $\mathbf{q}^m_{\text{Anchor}} = (x, y, z, d) \in \mathbb{R}^4$. As is standard practice, the corresponding *content queries* $\{\mathbf{q}^m_{\text{Content}}\}_{m=1}^M$, where each $\mathbf{q}^m_{\text{Content}} \in \mathbb{R}^C$, are initialized as learnable parameters.

3D Decoder. The 3D decoder iteratively updates the content queries and 3D anchors (Fig. 4d). Inspired by GroundingDINO [33], we employ a 3-stage attention block repeated L_{D} times. First, self-attention is applied to the content queries, followed by cross-attention between these queries

Table 1: **Zero-shot intra-instance detection results on the DS-10440 dataset.** The best-performing cryo-ET method for each metric is highlighted in bold. FullTilt achieves the best performance across all evaluated metrics, with the shortest runtime and lowest VRAM usage among cryo-ET methods.

Category	Method	# Prompts	mAP@0.5r ↑	mAP@1r ↑	F1 ↑	T. ↓	M. ↓
General MV	DETR3D [19]+$\mathcal{E}_{\text{Prompt}}$	1	0.048 ± 0.004	0.091 ± 0.005	0.146 ± 0.006	0.214	3430
	PETR [20]+$\mathcal{E}_{\text{Prompt}}$		0.000 ± 0.000	0.001 ± 0.000	0.010 ± 0.000	0.571	13800
Cryo-ET	Zeng et al. [22]+$\mathcal{E}_{\text{Prompt}}$		n.a.	n.a.	0.004 ± 0.001	71.3	4050
	TomoTwin [9]		0.089 ± 0.021	0.132 ± 0.024	0.078 ± 0.012	2960	20800
	CryoSAM [10]		n.a.	n.a.	0.008 ± 0.003	309	30600
	ProPicker [11]		n.a.	n.a.	0.061 ± 0.008	79.5	12700
	FullTilt (ours)		**0.118 ± 0.013**	**0.148 ± 0.013**	**0.193 ± 0.008**	**0.499**	**3430**
General MV	DETR3D [19]+$\mathcal{E}_{\text{Prompt}}$	2	0.056 ± 0.005	0.104 ± 0.005	0.161 ± 0.006	0.214	3430
	PETR [20]+$\mathcal{E}_{\text{Prompt}}$		0.000 ± 0.000	0.001 ± 0.000	0.010 ± 0.000	0.572	13800
Cryo-ET	Zeng et al. [22]+$\mathcal{E}_{\text{Prompt}}$		n.a.	n.a.	0.004 ± 0.001	71.3	4050
	TomoTwin [9]		0.066 ± 0.012	0.098 ± 0.015	0.089 ± 0.010	2960	20800
	CryoSAM [10]		n.a.	n.a.	0.008 ± 0.003	365	30600
	ProPicker [11]		n.a.	n.a.	0.072 ± 0.004	79.5	12700
	FullTilt (ours)		**0.129 ± 0.010**	**0.161 ± 0.011**	**0.209 ± 0.013**	**0.503**	**3430**
General MV	DETR3D [19]+$\mathcal{E}_{\text{Prompt}}$	4	0.062 ± 0.003	0.111 ± 0.003	0.168 ± 0.005	0.215	3430
	PETR [20]+$\mathcal{E}_{\text{Prompt}}$		0.000 ± 0.000	0.001 ± 0.000	0.010 ± 0.000	0.572	13800
Cryo-ET	Zeng et al. [22]+$\mathcal{E}_{\text{Prompt}}$		n.a.	n.a.	0.004 ± 0.001	71.3	4050
	TomoTwin [9]		0.067 ± 0.010	0.105 ± 0.015	0.102 ± 0.021	2960	20800
	CryoSAM [10]		n.a.	n.a.	0.043 ± 0.003	352	30600
	ProPicker [11]		n.a.	n.a.	0.081 ± 0.004	79.2	12700
	FullTilt (ours)		**0.133 ± 0.007**	**0.180 ± 0.007**	**0.221 ± 0.008**	**0.510**	**3430**

$+\mathcal{E}_{\text{Prompt}}$: with our multiclass visual prompt encoder module; MV: multi-view; n.a.: not applicable because the metric cannot be measured; T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

and the class prototypes. To perform the third stage—deformable cross-attention with the 3D-aware features—we duplicate the content queries for each tilt image and project their 3D anchors (x, y, z) into 2D coordinates $(\tilde{x}, \tilde{y})$ given the tilt angle θ: $\tilde{x} = (x - \frac{W}{2})\cos\theta + (z - \frac{D}{2})\sin\theta + \frac{W}{2}$ and $\tilde{y} = y$, where D is the tomogram depth. After computing deformable cross-attention using these projected coordinates, the output content queries are averaged across all tilt images. The 3D anchors are refined iteratively via a "look forward twice" approach [34], which propagates gradients from the next decoder layer through the current layer's anchor refinement to stabilize training and accelerate convergence. Ultimately, for each $m \in \{1, \ldots, M\}$, we take the final refined 3D anchor as the spatial estimation $\hat{\mathbf{P}}^m = (\hat{x}, \hat{y}, \hat{z}, \hat{d})$. The estimated class label $\hat{Y}^m$ and confidence score s^m are derived from the inner product similarity between the final content query and the prototypes, computed as $\mathbf{q}^m_{\text{Content}}\mathbf{t}^\top \in \mathbb{R}^{N_c}$. More architecture details are in Appendix B.

3.4 Training Objective

Following standard DETR architectures [32, 33], we compute an L1 loss for bounding box regression and a focal loss for classification, applied after bipartite matching between model outputs and ground truth. To accelerate convergence, we incorporate auxiliary losses at each decoder layer [33] and employ a contrastive denoising strategy [34], which additionally suppresses duplicate detections.

3.5 Auxiliary Geometric Primitives ($\mathcal{G}$)

Learning multi-view geometry directly from 2D tilt-series is exceptionally challenging, especially given adverse imaging artifacts (Fig. 3). As physics-based cryo-ET simulations are computationally prohibitive and scale poorly, we introduce an auxiliary geometric primitives module ($\mathcal{G}$). This module generates synthetic 2D projections (e.g., circles of varying sizes) alongside their exact 3D ground truth coordinates *on-the-fly*, eliminating I/O storage bottlenecks. $\mathcal{G}$ simulates diverse object distributions (e.g., dense clusters in the central z-plane) and mimics adverse imaging artifacts such as physical occlusion and non-uniform illumination. During training, we alternate between this dynamically generated data and the simulated cryo-ET data at every epoch. Details are provided in Appendix C.

3.6 Inference and Interaction

FullTilt's flexible design supports multiple interactive workflows. In a standard setup, a user selects a few target particles (e.g., 1–4 per class) from a reconstructed 3D tomogram to serve as prompts. For high-throughput analysis, users can extract class prototypes from one tomogram and reuse them across an entire dataset, enabling rapid zero-shot inference without manual re-prompting. In both cases, FullTilt automatically projects the 3D coordinates onto the 2D tilt images and seamlessly

Table 2: **Zero-shot intra-instance detection results on the EMPIAR-10304 dataset.** The best-performing cryo-ET method for each metric is highlighted in bold. FullTilt achieves the best detection performance except for mAP@0.5r in the 2- and 4-prompt settings, with the shortest runtime and lowest VRAM usage among cryo-ET methods.

Category	Method	# Prompts	mAP@0.5r ↑	mAP@1r ↑	F1 ↑	T. ↓	M. ↓
General MV	DETR3D [19]+$\mathcal{E}_{\text{Prompt}}$	1	0.011 ± 0.001	0.091 ± 0.002	0.222 ± 0.003	0.136	2290
	PETR [20]+$\mathcal{E}_{\text{Prompt}}$		0.000 ± 0.000	0.002 ± 0.000	0.032 ± 0.001	0.233	6020
Cryo-ET	Zeng et al. [22]+$\mathcal{E}_{\text{Prompt}}$		n.a.	n.a.	0.012 ± 0.001	30.2	2960
	TomoTwin [9]		0.189 ± 0.057	0.339 ± 0.063	0.532 ± 0.061	1730	20800
	CryoSAM [10]		n.a.	n.a.	0.002 ± 0.002	388	28400
	ProPicker [11]		n.a.	n.a.	0.224 ± 0.014	49.2	12500
	FullTilt (ours)		**0.199 ± 0.004**	**0.437 ± 0.002**	**0.593 ± 0.002**	**0.229**	**2410**
General MV	DETR3D [19]+$\mathcal{E}_{\text{Prompt}}$	2	0.012 ± 0.001	0.096 ± 0.003	0.227 ± 0.002	0.134	2290
	PETR [20]+$\mathcal{E}_{\text{Prompt}}$		0.000 ± 0.000	0.002 ± 0.000	0.032 ± 0.001	0.232	6020
Cryo-ET	Zeng et al. [22]+$\mathcal{E}_{\text{Prompt}}$		n.a.	n.a.	0.013 ± 0.001	30.2	2960
	TomoTwin [9]		**0.240 ± 0.046**	0.373 ± 0.050	0.569 ± 0.048	1720	20800
	CryoSAM [10]		n.a.	n.a.	0.004 ± 0.003	394	28400
	ProPicker [11]		n.a.	n.a.	0.239 ± 0.021	49.1	12500
	FullTilt (ours)		0.200 ± 0.004	**0.437 ± 0.003**	**0.592 ± 0.002**	**0.230**	**2410**
General MV	DETR3D [19]+$\mathcal{E}_{\text{Prompt}}$	4	0.012 ± 0.001	0.100 ± 0.003	0.230 ± 0.001	0.134	2290
	PETR [20]+$\mathcal{E}_{\text{Prompt}}$		0.000 ± 0.000	0.002 ± 0.000	0.032 ± 0.001	0.232	6020
Cryo-ET	Zeng et al. [22]+$\mathcal{E}_{\text{Prompt}}$		n.a.	n.a.	0.012 ± 0.001	30.2	2960
	TomoTwin [9]		**0.241 ± 0.022**	0.367 ± 0.026	0.565 ± 0.030	1720	20800
	CryoSAM [10]		n.a.	n.a.	0.008 ± 0.002	396	28400
	ProPicker [11]		n.a.	n.a.	0.247 ± 0.016	50.2	12500
	FullTilt (ours)		0.201 ± 0.004	**0.439 ± 0.001**	**0.593 ± 0.002**	**0.230**	**2410**

+$\mathcal{E}_{\text{Prompt}}$: with our multiclass visual prompt encoder module; MV: multi-view; n.a.: not applicable because the metric cannot be measured; T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

Table 3: **Zero-shot intra-instance detection results on the EMPIAR-10499 dataset.** The best-performing cryo-ET method for each metric is highlighted in bold. FullTilt achieves the best performance across all evaluated metrics, with the shortest runtime and lowest VRAM usage among cryo-ET methods.

Category	Method	# Prompts	mAP@0.5r ↑	mAP@1r ↑	F1 ↑	T. ↓	M. ↓
General MV	DETR3D [19]+$\mathcal{E}_{\text{Prompt}}$	1	0.039 ± 0.002	0.137 ± 0.004	0.271 ± 0.006	0.198	3230
	PETR [20]+$\mathcal{E}_{\text{Prompt}}$		0.000 ± 0.000	0.000 ± 0.000	0.017 ± 0.000	0.421	10600
Cryo-ET	Zeng et al. [22]+$\mathcal{E}_{\text{Prompt}}$		n.a.	n.a.	0.008 ± 0.001	26.9	4710
	TomoTwin [9]		0.035 ± 0.008	0.113 ± 0.020	0.249 ± 0.027	2330	20800
	CryoSAM [10]		n.a.	n.a.	0.032 ± 0.002	421	29200
	ProPicker [11]		n.a.	n.a.	0.187 ± 0.011	69.1	12700
	FullTilt (ours)		**0.159 ± 0.004**	**0.288 ± 0.004**	**0.418 ± 0.005**	**0.389**	**3410**
General MV	DETR3D [19]+$\mathcal{E}_{\text{Prompt}}$	2	0.049 ± 0.002	0.160 ± 0.004	0.299 ± 0.005	0.202	3230
	PETR [20]+$\mathcal{E}_{\text{Prompt}}$		0.000 ± 0.000	0.000 ± 0.000	0.017 ± 0.000	0.423	10600
Cryo-ET	Zeng et al. [22]+$\mathcal{E}_{\text{Prompt}}$		n.a.	n.a.	0.008 ± 0.001	26.8	4710
	TomoTwin [9]		0.057 ± 0.011	0.155 ± 0.023	0.296 ± 0.032	2320	20800
	CryoSAM [10]		n.a.	n.a.	0.034 ± 0.001	444	29200
	ProPicker [11]		n.a.	n.a.	0.206 ± 0.012	72.8	12700
	FullTilt (ours)		**0.169 ± 0.006**	**0.302 ± 0.009**	**0.432 ± 0.010**	**0.394**	**3410**
General MV	DETR3D [19]+$\mathcal{E}_{\text{Prompt}}$	4	0.056 ± 0.001	0.177 ± 0.002	0.317 ± 0.001	0.201	3230
	PETR [20]+$\mathcal{E}_{\text{Prompt}}$		0.000 ± 0.000	0.000 ± 0.000	0.016 ± 0.000	0.423	10600
Cryo-ET	Zeng et al. [22]+$\mathcal{E}_{\text{Prompt}}$		n.a.	n.a.	0.008 ± 0.001	26.9	4710
	TomoTwin [9]		0.078 ± 0.008	0.198 ± 0.016	0.354 ± 0.018	2310	20800
	CryoSAM [10]		n.a.	n.a.	0.038 ± 0.001	486	29200
	ProPicker [11]		n.a.	n.a.	0.235 ± 0.012	71.8	12700
	FullTilt (ours)		**0.173 ± 0.002**	**0.309 ± 0.003**	**0.441 ± 0.005**	**0.394**	**3410**

+$\mathcal{E}_{\text{Prompt}}$: with our multiclass visual prompt encoder module; MV: multi-view; n.a.: not applicable because the metric cannot be measured; T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

handles varying prompt quantities across views. In addition, users can successfully conduct full tilt-series inference by directly providing 2D prompts solely on the 0° tilt image.

4 Experiments

4.1 Datasets and Metrics

Training Dataset. To ensure fair comparisons with open-set methods [9, 11], we generate 1,138 simulated tomograms and corresponding aligned tilt-series (noisy and clean) using the same 119 proteins and established TEM simulation protocol from prior work [9, 35].

Evaluation Dataset. We evaluate FullTilt's zero-shot performance on three public real-world cryo-ET datasets: CZII CryoET Object Identification Challenge (DS-10440) [36], EMPIAR-10304 [37], and EMPIAR-10499 [38]. We additionally evaluate on a simulated dataset containing 10 unseen proteins alongside a training subset. Further dataset preparation details are provided in Appendix D.

Table 4: **Zero-shot cross-instance detection results on real-world cryo-ET datasets.** The best-performing open-set method for each metric is highlighted in bold. FullTilt outperforms all baselines, excepting mAP@0.5r under 2- and 4-prompt settings on EMPIAR-10304.

			DS-10440			EMPIAR-10304			EMPIAR-10499		
Cat.	**Method**	**# P.**	**mAP@0.5r ↑**	**mAP@1r ↑**	**F1 ↑**	**mAP@0.5r ↑**	**mAP@1r ↑**	**F1 ↑**	**mAP@0.5r ↑**	**mAP@1r ↑**	**F1 ↑**
WSL	Huang et al. [26]	1	-	-	-	-	-	-	-	-	-
	TomoPicker [12]		0.002 ± 0.001	0.035 ± 0.006	0.061 ± 0.010	0.027 ± 0.030	0.056 ± 0.038	0.205 ± 0.108	0.002 ± 0.001	0.017 ± 0.012	0.092 ± 0.055
OS	TomoTwin [9]		0.077 ± 0.026	0.121 ± 0.029	0.095 ± 0.026	0.203 ± 0.137	0.348 ± 0.189	0.508 ± 0.214	0.049 ± 0.051	0.164 ± 0.136	0.265 ± 0.196
	ProPicker [11]		n.a.	n.a.	0.076 ± 0.013	n.a.	n.a.	0.221 ± 0.071	n.a.	n.a.	0.196 ± 0.108
	FullTilt (ours)		**0.112 ± 0.033**	**0.137 ± 0.036**	**0.167 ± 0.037**	**0.204 ± 0.006**	**0.439 ± 0.005**	**0.593 ± 0.004**	**0.149 ± 0.051**	**0.271 ± 0.083**	**0.394 ± 0.134**
WSL	Huang et al. [26]	2	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
	TomoPicker [12]		0.002 ± 0.001	0.033 ± 0.009	0.055 ± 0.013	0.100 ± 0.131	0.149 ± 0.141	0.321 ± 0.188	0.003 ± 0.001	0.039 ± 0.019	0.159 ± 0.044
OS	TomoTwin [9]		0.091 ± 0.044	0.140 ± 0.057	0.078 ± 0.042	**0.216 ± 0.152**	0.336 ± 0.177	0.503 ± 0.206	0.055 ± 0.045	0.143 ± 0.105	0.265 ± 0.158
	ProPicker [11]		n.a.	n.a.	0.068 ± 0.018	n.a.	n.a.	0.218 ± 0.050	n.a.	n.a.	0.220 ± 0.109
	FullTilt (ours)		**0.124 ± 0.020**	**0.154 ± 0.022**	**0.186 ± 0.025**	0.199 ± 0.003	**0.437 ± 0.003**	**0.591 ± 0.002**	**0.147 ± 0.039**	**0.273 ± 0.062**	**0.401 ± 0.091**
WSL	Huang et al. [26]	4	0.000 ± 0.000	0.000 ± 0.000	0.004 ± 0.002	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
	TomoPicker [12]		0.006 ± 0.001	0.042 ± 0.004	0.080 ± 0.011	0.074 ± 0.054	0.215 ± 0.117	0.421 ± 0.127	0.018 ± 0.007	0.107 ± 0.020	0.256 ± 0.012
OS	TomoTwin [9]		0.106 ± 0.026	0.160 ± 0.038	0.111 ± 0.061	**0.238 ± 0.167**	0.357 ± 0.211	0.506 ± 0.262	0.064 ± 0.066	0.163 ± 0.123	0.272 ± 0.186
	ProPicker [11]		n.a.	n.a.	0.083 ± 0.011	n.a.	n.a.	0.249 ± 0.035	n.a.	n.a.	0.191 ± 0.079
	FullTilt (ours)		**0.136 ± 0.009**	**0.169 ± 0.009**	**0.200 ± 0.013**	0.200 ± 0.008	**0.437 ± 0.006**	**0.591 ± 0.004**	**0.177 ± 0.019**	**0.316 ± 0.023**	**0.456 ± 0.022**

Cat.: category; # P.: # prompts; WSL: weakly-supervised learning; OS: open-set; -: Huang et al. needs at least 2 prompts; n.a.: not applicable because the metric cannot be measured; ↓: lower is better; ↑: higher is better

Metrics. We report mean Average Precision (mAP) at distance thresholds of 0.5 and 1.0 times the target radius (mAP@0.5r, mAP@1.0r) [26]. As some baselines lack confidence scores, we supplement this with an F1 score (1.0 radius threshold). In all cases, inference runtime (T.) and peak VRAM (M.) are measured on a single NVIDIA RTX 6000 Ada GPU and 48-core CPU. Further details regarding the benchmark settings are provided in Appendix E.

4.2 Implementation Details

Following previous work [34, 33, 32], we use the Swin Transformer [17] as our 2D backbone, setting L_{AA}, L_{VP}, and L_{D} to 6, C to 256, and M to 900. FullTilt is implemented in PyTorch 2.5.1 and trained for 200 epochs via AdamW [39] (learning rate 1×10^{-4}). Training is distributed across four NVIDIA RTX 6000 Ada GPUs with a per-GPU batch size of 1. Further details regarding training and baseline implementations are provided in Appendix F and Appendix G, respectively.

4.3 Intra-Instance Detection

For intra-instance detection (i.e., prompts and targets reside in the same instance), we evaluate FullTilt against zero-shot open-set methods (TomoTwin [9], CryoSAM [10], ProPicker [11]). To benchmark against 2D back-projection (Zeng's method [22]) and general multi-view frameworks (DETR3D [19], PETR [20]), we integrated our prompt module ($\mathcal{E}_{\text{Prompt}}$) into their architectures. Fully supervised methods are omitted due to our focus on zero-shot detection and the absence of separate training sets.

We evaluate 1, 2, and 4 prompts over 10 independent trials (Tables 1–3; extended details and qualitative results in Appendix H.1 and Appendix I). Expectedly, providing more prompts generally enhances performance (except EMPIAR-10304). FullTilt achieves comparable or superior real-world performance across all metrics, reducing inference to under one second with the lowest VRAM requirements. Notably, FullTilt's margin over TomoTwin grows on challenging native cellular (EMPIAR-10499) versus purified (EMPIAR-10304) ribosomes. Among the baselines, TomoTwin outperforms ProPicker but incurs significantly longer runtimes, whereas CryoSAM struggles because its natural-image pre-training translates poorly to cryo-ET [11]. Despite being trained on identical data, general multi-view models (DETR3D, PETR) fail to effectively fuse tilt-series information. PETR's particularly poor performance indicates that attention across all tilts in the decoder cannot capture critical features.

4.4 Cross-Instance Detection

For cross-instance detection, a prompt from one instance is used for dataset-wide inference. We compare FullTilt against the top open-set baselines (TomoTwin [9], ProPicker [11]) and WSL methods (Huang's method [26], TomoPicker [12]). We randomly select one instance per dataset to extract prototype embeddings or train WSL models, using 1, 2, or 4 prompts. Experiments span 10 trials (limited to 3 for WSL models due to extensive training times). As shown in Table 4, FullTilt outperforms baselines across all real-world datasets, except for a marginally lower mAP@0.5r than

TomoTwin on EMPIAR-10304. As expected, WSL methods fail to generalize effectively with so few prompts, though TomoPicker surpasses Huang's method (details provided in Appendix H.2).

4.5 Multiclass Prompting Detection

Finally, we evaluate $\mathcal{E}_{\text{Prompt}}$ in multiclass scenarios for both intra- and cross-instance settings using the 6-class DS-10440 dataset. We test all possible class combinations over 10 independent trials (Table 5 and Appendix H.3). Although querying more classes marginally compromises detection performance, FullTilt processes all requested classes simultaneously with negligible runtime and VRAM overhead.

4.6 Ablation Studies

Number of Tilt Images vs. Projections per Prompt. We evaluate FullTilt under two distinct data reduction scenarios (Table 6). First, retaining fewer tilt images (by progressively doubling the angle increment) proportionally degrades detection performance, though it correspondingly improves runtime and VRAM usage. Conversely, supplying visual prompts in only a sparse subset of images, down to an extreme "single-image prompt" setting, only slightly compromises performance. More ablation experiments are provided in Appendix J.

Table 5: **Multi-class prompting performance comparison (intra-instance).** FullTilt maintains stable computational efficiency regardless of the number of target classes.

	1 Prompt				2 Prompts				4 Prompts			
# Classes	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**
1	0.118 ± 0.013	0.148 ± 0.013	0.499	3430	0.129 ± 0.010	0.161 ± 0.011	0.503	3430	0.133 ± 0.007	0.180 ± 0.007	0.510	3430
2	0.112 ± 0.005	0.142 ± 0.004	0.518	3430	0.123 ± 0.003	0.155 ± 0.003	0.522	3430	0.127 ± 0.003	0.161 ± 0.003	0.523	3430
3	0.111 ± 0.004	0.142 ± 0.004	0.521	3440	0.119 ± 0.002	0.151 ± 0.002	0.524	3440	0.123 ± 0.001	0.156 ± 0.001	0.525	3440
4	0.111 ± 0.002	0.142 ± 0.003	0.505	3440	0.118 ± 0.002	0.151 ± 0.001	0.507	3440	0.120 ± 0.001	0.153 ± 0.001	0.510	3440
5	0.108 ± 0.003	0.139 ± 0.004	0.514	3440	0.118 ± 0.002	0.150 ± 0.002	0.521	3440	0.119 ± 0.001	0.150 ± 0.002	0.524	3440
6	0.113 ± 0.008	0.144 ± 0.007	0.508	3440	0.115 ± 0.006	0.147 ± 0.006	0.507	3440	0.117 ± 0.003	0.149 ± 0.003	0.513	3440

T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

Table 6: **Ablation study on the number of tilt images and the number of projections per prompt.** Reducing the number of tilt images decreases runtime and memory at the cost of overall detection performance; reducing the number of projections per prompt only slightly compromises performance.

	DS-10440				EMPIAR-10304				EMPIAR-10499			
Configuration	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**
Baseline (100%)	0.118 ± 0.013	0.148 ± 0.013	0.499	3430	0.199 ± 0.004	0.437 ± 0.002	0.229	2410	0.159 ± 0.004	0.288 ± 0.004	0.389	3410
Ablation: Reduced Tilt Images												
50%	0.100 ± 0.013	0.133 ± 0.011	0.246	1860	0.184 ± 0.002	0.427 ± 0.002	0.118	1380	0.127 ± 0.006	0.260 ± 0.007	0.202	1950
25%	0.085 ± 0.010	0.120 ± 0.011	0.153	1280	0.157 ± 0.003	0.391 ± 0.003	0.079	860	0.060 ± 0.005	0.165 ± 0.008	0.116	1220
12.5%	0.038 ± 0.007	0.073 ± 0.014	0.087	700	0.084 ± 0.003	0.268 ± 0.003	0.069	600	0.010 ± 0.001	0.048 ± 0.003	0.083	790
Ablation: Reduced Projections per Prompt												
50%	0.111 ± 0.008	0.142 ± 0.009	0.514	3430	0.199 ± 0.003	0.437 ± 0.002	0.223	2420	0.162 ± 0.005	0.290 ± 0.005	0.392	3410
25%	0.107 ± 0.013	0.136 ± 0.014	0.513	3430	0.199 ± 0.004	0.437 ± 0.003	0.223	2420	0.158 ± 0.006	0.285 ± 0.007	0.391	3410
12.5%	0.108 ± 0.009	0.137 ± 0.010	0.512	3430	0.197 ± 0.005	0.436 ± 0.003	0.224	2420	0.158 ± 0.005	0.285 ± 0.006	0.391	3410
Only One	0.106 ± 0.014	0.134 ± 0.016	0.513	3430	0.190 ± 0.004	0.430 ± 0.004	0.224	2420	0.155 ± 0.009	0.278 ± 0.011	0.390	3410

T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

Table 7: **Ablation study on FullTilt components.** We ablate the tilt-series encoder ($\mathcal{E}_{\text{Tilt}}$), tilt-aware query initializer ($\mathcal{I}_{\text{Tilt}}$), and auxiliary geometric primitives ($\mathcal{G}$) on detection performance across three real-world datasets. The integration of all three modules yields the optimal performance.

			DS-10440				EMPIAR-10304				EMPIAR-10499			
$\mathcal{E}_{\text{Tilt}}$	$\mathcal{I}_{\text{Tilt}}$	$\mathcal{G}$	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**
			0.001 ± 0.001	0.003 ± 0.001	0.215	3430	0.000 ± 0.000	0.001 ± 0.000	0.133	2280	0.000 ± 0.000	0.000 ± 0.000	0.200	3170
✓			0.002 ± 0.001	0.008 ± 0.001	0.519	3430	0.000 ± 0.000	0.005 ± 0.001	0.229	2300	0.000 ± 0.000	0.002 ± 0.000	0.380	3190
	✓		0.040 ± 0.002	0.099 ± 0.003	0.223	3430	0.004 ± 0.000	0.106 ± 0.003	0.141	2400	0.009 ± 0.001	0.082 ± 0.003	0.207	3380
		✓	0.025 ± 0.002	0.063 ± 0.003	0.211	3430	0.000 ± 0.000	0.016 ± 0.001	0.139	2280	0.001 ± 0.000	0.011 ± 0.001	0.198	3170
✓	✓		0.099 ± 0.007	0.131 ± 0.008	0.517	3430	0.008 ± 0.001	0.136 ± 0.009	0.228	2420	0.049 ± 0.003	0.168 ± 0.006	0.392	3390
	✓	✓	0.094 ± 0.004	0.153 ± 0.005	0.222	3430	0.030 ± 0.001	0.181 ± 0.003	0.142	2400	0.033 ± 0.001	0.127 ± 0.003	0.207	3380
✓		✓	0.002 ± 0.001	0.008 ± 0.002	0.520	3430	0.001 ± 0.000	0.017 ± 0.001	0.229	2300	0.000 ± 0.000	0.004 ± 0.000	0.379	3190
✓	✓	✓	0.118 ± 0.013	0.148 ± 0.013	0.499	3430	0.199 ± 0.004	0.437 ± 0.002	0.229	2410	0.159 ± 0.004	0.288 ± 0.004	0.389	3410

✓: has this feature; $\mathcal{E}_{\text{Tilt}}$: tilt-series encoder; $\mathcal{I}_{\text{Tilt}}$: tilt-aware query initializer; $\mathcal{G}$: auxiliary geometric primitives; T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

Component Ablation. We ablate our proposed modules: the tilt-series encoder ($\mathcal{E}_{\text{Tilt}}$), tilt-aware query initializer ($\mathcal{I}_{\text{Tilt}}$), and auxiliary geometric primitives module ($\mathcal{G}$) (Table 7; granular design ablations in Appendix J.5 and Appendix J.6). Without these modules, FullTilt fails. Among single-module configurations, only $\mathcal{I}_{\text{Tilt}}$ yields viable detections. Integrating $\mathcal{E}_{\text{Tilt}}$ subsequently drives a dramatic performance surge, underscoring the absolute necessity of cross-tilt fusion for cryo-ET data. Finally, adding $\mathcal{G}$ further boosts performance by enhancing the model's understanding of multi-view geometry and improving its robustness to adverse imaging artifacts. Overall, integrating all three modules yields the best performance across the evaluated metrics, except for mAP@1r on DS-10440.

5 Conclusion

In this work, we introduced FullTilt, an end-to-end framework that redefines open-set 3D macromolecule detection by operating directly on aligned 2D cryo-ET tilt-series. By bypassing computationally expensive 3D sliding-window inference, FullTilt eliminates redundant volumetric computation, reducing runtime by orders of magnitude while strictly limiting VRAM usage. Extensive evaluations demonstrate that FullTilt achieves state-of-the-art zero-shot detection performance on real-world datasets. We anticipate that demonstrating the efficiency and effectiveness of this direct 2D-to-3D detection paradigm will facilitate rapid, large-scale object detection in structural biology and beyond.

Limitations and Future Work. FullTilt currently struggles with particularly small macromolecules, with near-zero AP for all evaluated protein types below 100 kDa (see Appendix H.6), consistent with the small-object detection challenges of DETR-like architectures [16]. Its relatively weaker performance on simulated data may arise because FullTilt and competing methods overfit to different simulation-specific cues, such as particle-placement patterns and noise statistics. Our preliminary experiments suggest that broader training-data coverage can improve FullTilt's performance on simulated data (see Appendix H.7); future work will further investigate how reliance on these cues affects generalization. Moreover, zero-shot performance remains below that of fully supervised methods, potentially due to the substantial domain gap between simulated and real-world data. This motivates real-world fine-tuning and further large-scale training, for which FullTilt can serve as a foundation. Future work will also integrate our macromolecule detection framework into downstream subtomogram averaging [4].

Acknowledgements

Research reported in this publication was supported by the National Institute Of General Medical Sciences of the National Institutes of Health under Award Number R35GM163737. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. M.-Y.H. also acknowledges support from the Hung Taiwan-Duke University Fellowship. This study utilized the computational resources offered by Duke Research Computing (http://rc.duke.edu). We thank Tracy Futhey, Charley Kneifel, Katie Kilroy, Mike Newton, Victor Orlikowski, Tom Milledge, and David Lane from the Duke Office of Information Technology and Research Computing for providing assistance with the computing environment.

References

[1] T. L. Blundell, "Structure-based drug design," *Nature*, vol. 384, no. 6604, p. 23, 1996. 1

[2] M. Duran-Frigola, R. Mosca, and P. Aloy, "Structural systems pharmacology: the role of 3d structures in next-generation drug development," *Chemistry & biology*, vol. 20, no. 5, pp. 674–684, 2013. 1

[3] M. Turk and W. Baumeister, "The promise and the challenges of cryo-electron tomography," *FEBS letters*, vol. 594, no. 20, pp. 3243–3261, 2020. 1

[4] A. J. Watson and A. Bartesaghi, "Advances in cryo-et data processing: meeting the demands of visual proteomics," *Current Opinion in Structural Biology*, vol. 87, p. 102861, 2024. 1, 10

[5] T. Wagner and S. Raunser, "Cryo-electron tomography: Challenges and computational strategies for particle picking," *Current Opinion in Structural Biology*, vol. 93, p. 103113, 2025. 1

[6] T. Wagner, F. Merino, M. Stabrin, T. Moriya, C. Antoni, A. Apelbaum, P. Hagel, O. Sitsel, T. Raisch, D. Prumbaum, *et al.*, "Sphire-cryolo is a fast and accurate fully automated particle picker for cryo-em," *Communications biology*, vol. 2, no. 1, p. 218, 2019. 1, 2

[7] 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*, vol. 18, no. 11, pp. 1386–1394, 2021. 2

[8] 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*, vol. 15, no. 1, p. 2090, 2024. 1

[9] 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*, vol. 20, no. 6, pp. 871–880, 2023. 1, 2, 3, 6, 7, 8, 18, 19, 21, 23, 24, 27, 34

[10] Y. Zhao, H. Bian, M. Mu, M. R. Uddin, Z. Li, X. Li, T. Wang, and M. Xu, "Cryosam: Training-free cryoet tomogram segmentation with foundation models," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 124–134, Springer, 2024. 1, 2, 3, 6, 7, 8, 18, 21, 23, 24

[11] S. Wiedemann, Z. Fabian, M. Soltanolkotabi, and R. Heckel, "Propicker: promptable segmentation for particle picking in cryogenic electron tomography," *Journal of Structural Biology*, p. 108298, 2026. 1, 2, 3, 6, 7, 8, 18, 21, 23, 24, 27, 34

[12] M. R. Uddin, A. Y. Ahmed, M. T. Tahmid, M. Z. U. Alam, Z. Freyberg, and M. Xu, "Tomopicker: annotation-efficient particle picking in cryo-electron tomograms," *bioRxiv*, 2024. 1, 2, 8, 21, 27

[13] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023. 1, 3, 21

[14] M. Levoy, *Volume rendering using the Fourier projection-slice theorem*. Computer Systems Laboratory, Stanford University, 1992. 2

[15] H. Winkler and K. A. Taylor, "Accurate marker-free alignment with simultaneous geometry determination and reconstruction of tilt series in electron tomography," *Ultramicroscopy*, vol. 106, no. 3, pp. 240–254, 2006. 2

[16] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*, pp. 213–229, Springer, 2020. 2, 3, 4, 10, 14, 15

[17] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021. 2, 8

[18] A. Singh, "Surround-view vision-based 3d detection for autonomous driving: A survey," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3243–3252, 2023. 2

[19] Y. Wang, V. C. Guizilini, T. Zhang, Y. Wang, H. Zhao, and J. Solomon, "Detr3d: 3d object detection from multi-view images via 3d-to-2d queries," in *Conference on robot learning*, pp. 180–191, PMLR, 2022. 2, 3, 6, 7, 8, 14, 22, 23, 24

[20] Y. Liu, T. Wang, X. Zhang, and J. Sun, "Petr: Position embedding transformation for multi-view 3d object detection," in *European conference on computer vision*, pp. 531–548, Springer, 2022. 2, 3, 6, 7, 8, 23, 24

[21] Z. Yang, D. Zang, H. Li, Z. Zhang, F. Zhang, and R. Han, "Self-supervised noise modeling and sparsity guided electron tomography volumetric image denoising," *Ultramicroscopy*, vol. 255, p. 113860, 2024. 2

[22] X. Zeng, Z. Lin, M. R. Uddin, B. Zhou, C. Cheng, J. Zhang, Z. Freyberg, and M. Xu, "Structure detection in three-dimensional cellular cryoelectron tomograms by reconstructing two-dimensional annotated tilt series," *Journal of Computational Biology*, vol. 29, no. 8, pp. 932–941, 2022. 2, 3, 6, 7, 8, 21, 23, 24

[23] E. Genthe, S. Miletic, I. Tekkali, R. H. James, T. C. Marlovits, and P. Heuser, "Pickyolo: Fast deep learning particle detector for annotation of cryo electron tomograms," *Journal of structural biology*, vol. 215, no. 3, p. 107990, 2023. 2

[24] I. de Teresa-Trueba, S. K. Goetz, A. Mattausch, F. Stojanovska, C. E. Zimmerli, M. Toro-Nahuelpan, D. W. Cheng, F. Tollervey, C. Pape, M. Beck, *et al.*, "Convolutional networks for supervised mining of molecular patterns within cellular context," *Nature Methods*, vol. 20, no. 2, pp. 284–294, 2023. 2

[25] Y. Hao, B. Zhang, X. Wan, R. Yan, Z. Liu, J. Li, S. Zhang, X. Cui, and F. Zhang, "Vp-detector: A 3d convolutional neural network for automated macromolecule localization and classification in cryo-electron tomograms," *bioRxiv*, pp. 2021–05, 2021. 2

[26] Q. Huang, Y. Zhou, H.-F. Liu, and A. Bartesaghi, "Accurate detection of proteins in cryo-electron tomograms from sparse labels," in *European Conference on Computer Vision*, pp. 644–660, Springer, 2022. 2, 8, 18, 21, 27

[27] R. Kiryo, G. Niu, M. C. Du Plessis, and M. Sugiyama, "Positive-unlabeled learning with non-negative risk estimator," *Advances in neural information processing systems*, vol. 30, 2017. 2

[28] M. Alzahrani, M. Usman, S. K. Jarraya, S. Anwar, and T. Helmy, "Deep models for multi-view 3d object recognition: a review," *Artificial Intelligence Review*, vol. 57, no. 12, p. 323, 2024. 3

[29] Y. Liu, J. Yan, F. Jia, S. Li, A. Gao, T. Wang, and X. Zhang, "Petrv2: A unified framework for 3d perception from multi-camera images," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3262–3272, 2023. 3

[30] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: deformable transformers for end-to-end object detection," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, OpenReview.net, 2021. 5, 15

[31] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny, "Vggt: Visual geometry grounded transformer," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025. 5, 14

[32] Q. Jiang, F. Li, Z. Zeng, T. Ren, S. Liu, and L. Zhang, "T-rex2: Towards generic object detection via text-visual prompt synergy," in *European Conference on Computer Vision*, pp. 38–57, Springer, 2024. 5, 6, 8, 21

[33] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," in *European conference on computer vision*, pp. 38–55, Springer, 2024. 5, 6, 8, 15, 21, 22, 23

[34] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "Dino: Detr with improved denoising anchor boxes for end-to-end object detection," 2022. 6, 8

[35] H. Rullgård, L.-G. Öfverstedt, S. Masich, B. Daneholt, and O. Öktem, "Simulation of transmission electron microscope images of biological specimens," *Journal of microscopy*, vol. 243, no. 3, pp. 234–256, 2011. 7, 18

[36] A. Peck, Y. Yu, J. Schwartz, A. Cheng, U. H. Ermel, J. Hutchings, S. Kandel, D. Kimanius, E. A. Montabana, D. Serwas, *et al.*, "A realistic phantom dataset for benchmarking cryo-et data annotation," *Nature Methods*, vol. 22, no. 9, pp. 1819–1823, 2025. 7, 18, 19

[37] F. Eisenstein, R. Danev, and M. Pilhofer, "Improved applicability and robustness of fast cryo-electron tomography data acquisition," *Journal of structural biology*, vol. 208, no. 2, pp. 107–114, 2019. 7, 18, 19

[38] D. Tegunov, L. Xue, C. Dienemann, P. Cramer, and J. Mahamid, "Multi-particle cryo-em refinement with m visualizes ribosome-antibiotic complex at 3.5 å in cells," *Nature methods*, vol. 18, no. 2, pp. 186–193, 2021. 7, 18, 19

[39] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019. 8, 21

[40] H.-F. Liu, Y. Zhou, Q. Huang, J. Piland, W. Jin, J. Mandel, X. Du, J. Martin, and A. Bartesaghi, "nextpyp: a comprehensive and scalable platform for characterizing protein variability in situ using single-particle cryo-electron tomography," *Nature Methods*, vol. 20, no. 12, pp. 1909–1919, 2023. 19

[41] A. Bartesaghi, D. Matthies, S. Banerjee, A. Merk, and S. Subramaniam, "Structure of β-galactosidase at 3.2-å resolution obtained by cryo-electron microscopy," *Proceedings of the National Academy of Sciences*, vol. 111, no. 32, pp. 11709–11714, 2014. 19

[42] S. Jain, X. Li, and M. Xu, "Knowledge transfer from macro-world to micro-world: enhancing 3d cryo-et classification through fine-tuning video-based deep models," *Bioinformatics*, vol. 40, no. 7, p. btae368, 2024. 19

[43] A. Wang, L. Liu, H. Chen, Z. Lin, J. Han, and G. Ding, "Yoloe: Real-time seeing anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 24591–24602, 2025. 21

[44] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016. 21

[45] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981. 22

[46] T. Dao, D. Fu, S. Ermon, A. Rudra, and C. Ré, "Flashattention: Fast and memory-efficient exact attention with io-awareness," *Advances in neural information processing systems*, vol. 35, pp. 16344–16359, 2022. 54

Appendix

Appendix A Comparison with DETR3D and PETR, and FullTilt Design Rationale

DETR3D [19] extends the DETR [16] framework for multi-view 3D object detection. In each decoder layer, it projects 3D anchors onto each 2D image to incorporate local features into the object queries. Conversely, PETR eliminates the explicit 3D-to-2D projection step by directly infusing 2D features with 3D spatial information via 3D positional encoding. In the PETR decoder layer, the 3D object queries perform multi-head attention over all flattened features to update their representations.

Both frameworks present distinct advantages and limitations. For DETR3D, the 3D-to-2D projection operation ensures training efficiency and stability by avoiding attention over all features across all images. However, this projection requires careful geometric engineering. PETR addresses the projection bottleneck by enabling features to inherently possess 3D positional awareness; however, performing attention across all features from all images is computationally expensive and often leads to unstable training.

This instability is significantly exacerbated in cryo-ET data. Unlike autonomous driving scenarios where a small number of large, distinguishable objects (e.g., cars) are present, cryo-ET data suffers from extremely low signal-to-noise ratios and contains high concentration of small particles that are difficult to distinguish in any single tilt image. In such densely populated, low-contrast environments, PETR's global attention mechanism (i.e., attention across all features from all images) struggles to accurately localize the exact same target across different tilts, even with 3D-aware features. Furthermore, while particles become identifiable when observing consecutive frames, neither DETR3D nor PETR incorporates a fusion module to integrate information across the entire tilt-series.

Our FullTilt architecture addresses these challenges by synthesizing the strengths of both frameworks while introducing critical, cryo-ET-tailored modules. In our tilt-series encoder, we introduce a tilt embedding that grants the 2D features 3D spatial awareness, analogous to PETR's approach. Moreover, we incorporate a local attention and global row attention to fuse information across all tilt images.

A naive implementation of this fusion would flatten all features from all tilt images and apply self-attention [31]. However, a standard tilt-series contains over 30 images, each potentially yielding more than 4,000 features from multi-level feature maps, rendering global self-attention computationally prohibitive. To overcome this, we exploit the geometric prior of cryo-ET data: because the tilt-series is aligned along the y-axis, a given particle will consistently appear at the same y-axis coordinate across all tilt images. By restricting our module to a global row attention mechanism, combined with local attention, we efficiently fuse multi-tilt information into the corresponding features.

Finally, to support open-set detection scenarios, FullTilt incorporates a multiclass visual prompt encoder. A tilt-aware query initializer is introduced to provide 3D anchor priors for the decoder to refine. Within the decoder, we adopt DETR3D's 3D-to-2D projection strategy to strictly attend to localized target particles, ensuring that our training remains both highly stable and computationally efficient.

Appendix B FullTilt Architecture Details

Appendix B.1 External and Internal Label Mapping

To accommodate arbitrary user-defined class identifiers ($Y \in \mathbb{Z}^+$)—including sparse, excessively large, or non-contiguous label sets—we implement a dynamic mapping mechanism. In the beginning, all unique *external* labels provided by the user are hashed and mapped to a contiguous, zero-indexed *internal* label space $(0, 1, \ldots, N_c - 1$, where N_c is the number of unique classes). This internal representation facilitates efficient tensor indexing and matrix gathering operations within the network pipeline. Following inference, the estimated internal class indices are deterministically mapped back to the user's original external labels.

Appendix B.2 Normalized Coordinate Space

Following standard DETR design principles [16, 30, 33], all 3D spatial coordinates within the pipeline are strictly normalized to the $[0, 1]$ range. Because the intermediate transformer layer outputs exist in unconstrained real space ($\mathbb{R}$), we utilize standard sigmoid (σ) and inverse sigmoid (σ^{-1}) transformations for stable coordinate refinement. For instance, to update a normalized 3D anchor using the decoder's content queries, we first apply σ^{-1} to the anchor, add the unconstrained output from the model, and then apply σ to project the updated coordinate back into the normalized $[0, 1]$ range. This normalization strategy fundamentally impacts the 3D-to-2D projection geometry, necessitating precise anisotropic scaling factors to account for the varying absolute dimensions of the volumetric space. Additionally, the voxel or pixel size becomes unequal across dimensions within this normalized space. To address this spatial distortion in practice, the decoder explicitly regresses the diameter d as independent height (h) and width (w) variables, deriving the final scalar diameter by averaging these two values. In the final step, all estimated coordinates are scaled back from the normalized space to the absolute spatial dimensions of the original 3D tomogram.

Appendix C Auxiliary Geometric Primitives ($\mathcal{G}$) Implementation Details

The auxiliary geometric primitives module ($\mathcal{G}$) has two primary objectives: (1) to enable FullTilt to learn multi-view geometry without relying exclusively on computationally expensive simulated cryo-ET data, which cannot be generated *on-the-fly* and incurs significant storage overhead (typically over 2 GB per simulation); and (2) to improve the model's robustness to adverse imaging artifacts, such as severe physical occlusion and non-uniform illumination. One example of the resulting synthetic tilt-series is provided in Figs. A1 and A2. The generation pipeline, implemented entirely in `NumPy`, synthesizes a complete tilt-series of varying dimensions and angular increments in approximately 0.5 to 1.0 seconds.

Parameter Sampling and Coordinate Generation. The process begins by *empirically* sampling the spatial dimensions from discrete ($\mathcal{U}_{\text{Int}}$) or continuous ($\mathcal{U}$) uniform distributions: image height $H \sim \mathcal{U}_{\text{Int}}(300, 512)$, width $W \sim \mathcal{U}_{\text{Int}}(300, 512)$, and tomogram depth $D \sim \mathcal{U}_{\text{Int}}(120, 300)$. To control particle density along the z-axis, we define a central bounding slab of depth $D' \sim \mathcal{U}_{\text{Int}}(80, 120)$, ensuring $D' \leq D$. We then uniformly sample $M \sim \mathcal{U}_{\text{Int}}(100, 600)$ 3D coordinates within this D' slab. The particle diameter $d \sim \mathcal{U}_{\text{Int}}(6, 16)$ and opacity $c \sim \mathcal{U}(0.03, 0.10)$ are also randomly assigned. Additional parameters governing the illumination gradients, Gaussian noise, and boundary occlusions are drawn from empirical distributions designed to cover the variance observed in real-world cryo-ET datasets.

Projection and Visibility Masking. After selecting a random angular range and tilt increment, the 3D coordinates are projected onto each 2D plane. To enforce strict cross-view geometric consistency, any particle that is not fully visible across *all* generated tilt images is discarded.

Rendering and Adverse Condition Simulation. For each tilt angle, the valid projected coordinates are rendered as solid 2D disk kernels (circles). We then apply the sampled illumination gradient and Gaussian noise. Finally, we inject random left and right physical occlusions (simulating field-of-view truncation). These synthetic 2D images and their corresponding 3D ground truth coordinates are subsequently passed through our training augmentation pipeline (e.g., random resizing, padding, and flipping).

Limitations. While $\mathcal{G}$ effectively models multi-view geometry and core adverse imaging artifacts, incorporating a wider variety of non-spherical geometric primitives and complex physical conditions could further enrich the synthetic diversity; we leave this to future work. Additionally, due to the extensive computational time required for full model training, conducting an exhaustive hyperparameter search or an ablation study for $\mathcal{G}$ remains computationally prohibitive.

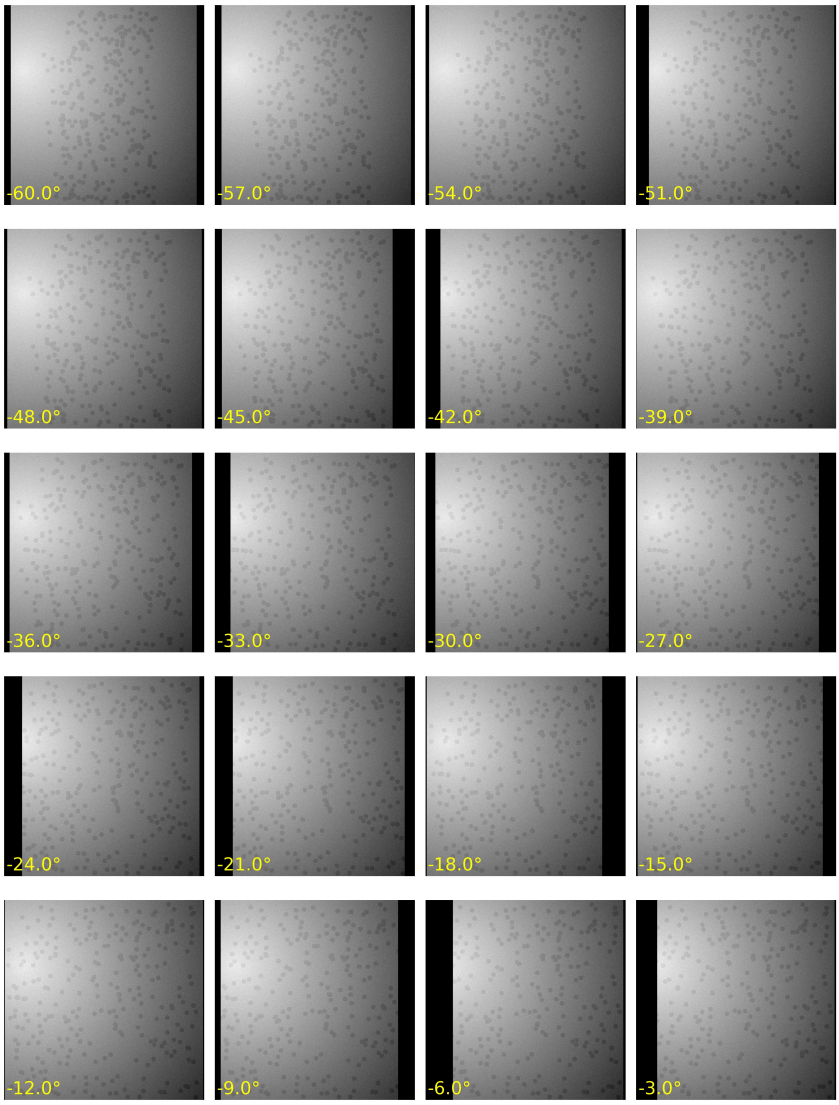


Figure A1: **A tilt-series generated by our auxiliary geometric primitives module.**

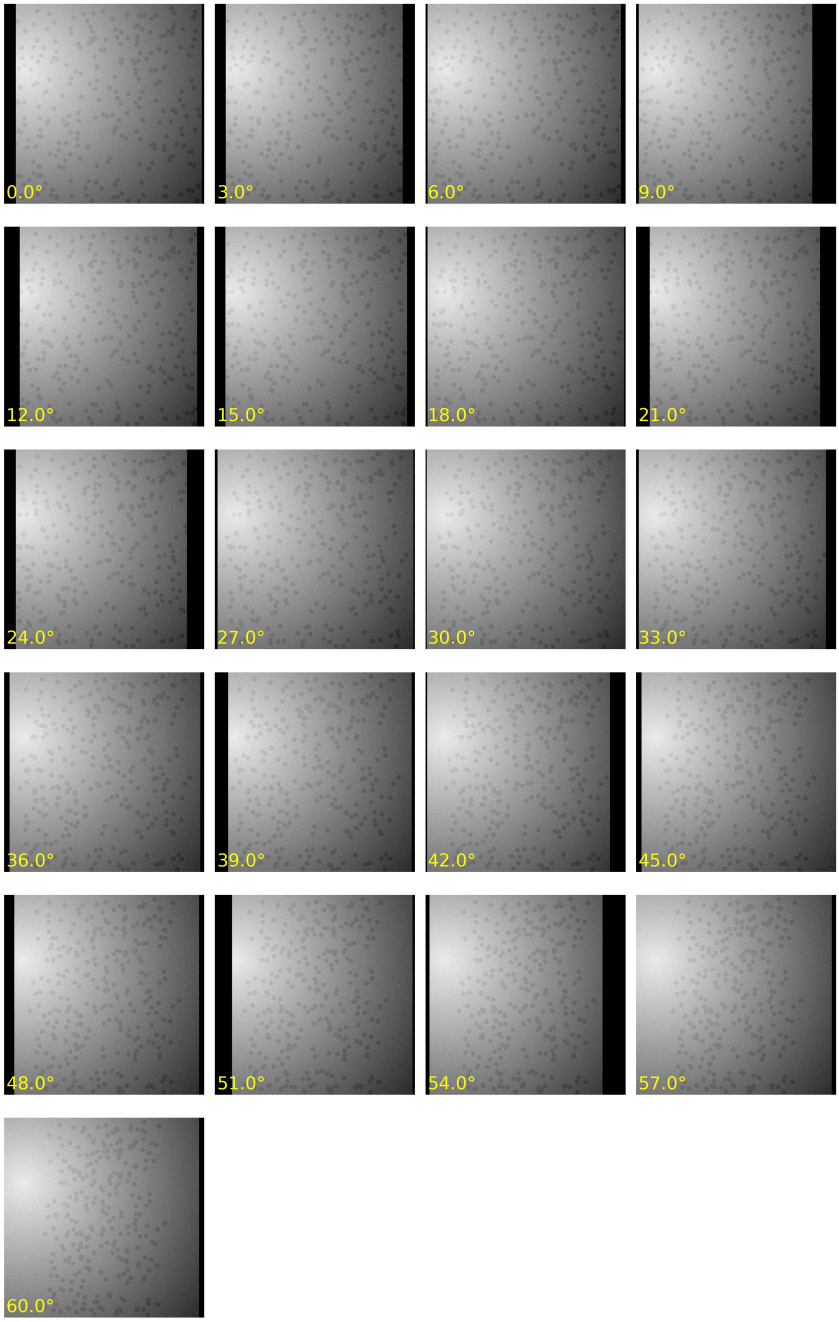


Figure A2: **A tilt-series generated by our auxiliary geometric primitives module (cont.).**

Appendix D Data Preparation Details

Appendix D.1 Training Dataset

Following the protocol established by TomoTwin [9], we utilize the TEM Simulator [35] to generate our simulated cryo-ET training dataset. Each generated instance comprises a 2D tilt-series, the ground truth 3D spatial coordinates of the particles, and a fully reconstructed 3D tomogram. Notably, our framework strictly utilizes the tilt-series and 3D coordinates, entirely discarding the reconstructed tomogram during training. To ensure a fair and direct comparison with state-of-the-art open-set 3D macromolecule detection methods (i.e., TomoTwin [9] and ProPicker [11]), we construct our simulations using the same core set of proteins. The complete list of the 119 proteins utilized in our study is provided in Table A1. Note that while the original TomoTwin protocol included 120 proteins, we omitted PDB entry `5MRC.pdb` due to download unavailability, resulting in our final count of 119.

The primary deviation from the TomoTwin simulation protocol lies in the distribution of particle classes and particle numbers per instance. We generated a diverse dataset consisting of 338 single-class, 300 two-class, 300 four-class, and 300 eight-class instances. For the single-class instances, we systematically cycled through the 119 proteins to guarantee each was represented at least twice, whereas the proteins for the multi-class instances were selected randomly. To control spatial crowding, the number of particles per protein was sampled randomly based on the total class count: 120–320 for single-class, 60–160 for two-class, 30–80 for four-class, and 15–40 for eight-class instances.

This specific compositional balance contrasts with TomoTwin's highly dense simulation settings (e.g., 150 particles distributed across 10 classes). Because subvolume-matching methods like TomoTwin extract small, localized 3D patches (e.g., $37 \times 37 \times 37$ voxels), generating densely populated tomograms is highly efficient, allowing them to harvest numerous positive subvolumes from a single simulation. In contrast, FullTilt solves a fundamentally different and highly complex multi-view geometry problem directly from multi-view projections. If the 3D volume is excessively dense, the corresponding 2D projections suffer from severe projection overlap and ambiguity. Therefore, our adjusted density distribution is specifically designed to mitigate extreme projection overlap, facilitating robust geometric learning and accurate 3D localization. We acknowledge this controlled density as a limitation of our current training paradigm and plan to address it in future work.

Table A1: **Training PDBs.**

1BXN	1G3I	1N9G	1QVR	1S3X	1SS8	1U6G	1UL1	2CG9	2DFS
2R9R	2RHS	2VZ9	2WW2	2XNX	3CF3	3D2F	3GL1	3H84	3LUE
3MKQ	3QM1	3ULV	4CR2	4UIC	4WRM	4XK8	5A20	5CSA	5G04
5H0S	5JH9	5LJO	5O32	5OOL	5VKQ	6AHU	6BQ1	6CES	6CNJ
6DUZ	6EMK	6F8L	6GY6	6GYM	6I0D	6ID1	6IGC	6JY0	6KLH
6KRK	6KSP	6LMT	6LX3	6LXK	6LXV	6M04	6MRC	6PIF	6SCJ
6TA5	6TAV	6TGC	6TPS	6U8Q	6UP6	6VGR	6VN1	6VZ8	6W6M
6WZT	6X5Z	6X9Q	6XF8	6YT5	6Z3A	6Z6O	6Z80	6ZIU	6ZQJ
7AMV	7B5S	7B7U	7BKC	7BLQ	7BLR	7DD9	7E1Y	7E6G	7E8H
7EEP	7EGD	7EGE	7EGQ	7ETM	7EY7	7JSN	7K5X	7KDV	7KFE
7KJ2	7LSY	7MEI	7NHS	7NIU	7NYZ	7O01	7Q21	7QJ0	7R04
7S7K	7SFW	7SGM	7SHK	7SN7	7T3U	7VTQ	7WBT	7WOO	

Appendix D.2 Evaluation Datasets

We evaluate FullTilt across a comprehensive suite of real-world and simulated cryo-ET datasets to rigorously assess its detection capabilities and generalization. We first benchmark our framework on the CZII CryoET Object Identification Challenge dataset (CryoET Data Portal Dataset ID: DS-10440) [36], as it serves as a premier, multi-class real-world benchmark with meticulously curated ground truth annotations. To further validate our method against established standards, we select EMPIAR-10304 [37] and EMPIAR-10499 [38] due to their widespread adoption in the literature [26, 10]. These datasets represent single-class scenarios, evaluating performance in both a purified (EMPIAR-10304) and a native cellular environment (EMPIAR-10499). Crucially, all real-world evaluations are conducted in a strictly zero-shot setting. However, acknowledging the inherent challenge of obtaining exhaustive, perfect ground truth for empirical EMPIAR data, we supplement our real-world benchmarks with fully annotated, multiclass simulation datasets. These simulations

provide exact 3D ground truth coordinates, enabling precise quantification of model performance in both "unseen" (zero-shot) and "seen" (non-zero-shot) scenarios. A statistical summary of all evaluation datasets is provided in Table A2.

Table A2: **Dataset Details.** Characteristics of the real-world and simulated Cryo-ET datasets used to evaluate FullTilt. Only the "seen" simulated cryo-ET dataset is not evaluated in a zero-shot setting.

	Real-World Cryo-ET Dataset			Simulated Cryo-ET Dataset	
	DS-10440 [36]	**EMPIAR-10304** [37]	**EMPIAR-10499** [38]	**Seen**	**Unseen**
# of Instances	7	12	65	20	20
# of Classes	6	1	1	10	10
# of Particles (per instance)	181.3 ± 35.2 [140, 225]	979.2 ± 72.6 [819, 1111]	382.1 ± 49.1 [257, 490]	100.0 ± 0.0 [100, 100]	100.0 ± 0.0 [100, 100]
# of Particles (per instance and per class)	30.2 ± 20.0 [9, 95]	979.2 ± 72.6 [819, 1111]	382.1 ± 49.1 [257, 490]	50.0 ± 30.6 [25, 100]	50.0 ± 30.6 [25, 100]
XY Pixel space (Å)	10	25.2	13.6	10.2	10.2
Tilt Range & Increment	−45° to +45° (3° step)	−60° to +60° (3° step)	−60° to +60° (3° step)	−60° to +60° (2° step)	−60° to +60° (2° step)
# of Projections	31	41	41	61	61
Projection Size (H×W)	630×630	480×340	464×480	512×512	512×512
Rec. Tomogram Size (D×H×W)	184×630×630	256×480×340	256×464×480	200×512×512	200×512×512

Rec.: reconstructed; # of particles is reported in mean ± std [min, max]

Appendix D.2.1 Zero-Shot Setting

DS-10440 [36]. We downloaded all seven raw tilt-series from the CryoET Data Portal (Deposition ID: CZCDP-10310; Dataset ID: DS-10440) and performed alignment according to the official protocol using the provided alignment files. Each tilt-series contains six distinct protein classes: apoferritin, β-amylase, β-galactosidase, ribosome, thyroglobulin, and virus-like particles. The corresponding 3D ground truth coordinates were obtained from the official Kaggle competition dataset.

EMPIAR-10304 [37]. We downloaded all 12 raw tilt-series from the EMPIAR archive and aligned them using IMOD. This dataset features a single target class: the purified *Escherichia coli* 70S ribosome. Due to the prohibitive difficulty of exhaustive manual annotation, 3D ground truth coordinates were first generated using an established size-based particle detection pipeline [40, 41] with optimized parameters, followed by meticulous manual examination to ensure high fidelity. While a small fraction of particles may remain unlabeled, these annotations are qualitatively robust for empirical benchmarking (see Fig. A14). Notably, our computational efficiency benchmarks (inference runtime and VRAM consumption) are entirely independent of label completeness. During inference, we omit the first and last five tilt images of each series due to severe adverse imaging artifacts (e.g., total physical occlusion).

EMPIAR-10499 [38]. We downloaded all 65 raw tilt-series from the EMPIAR archive, which were similarly aligned using IMOD. This dataset contains a single target class: the ribosome from native *Mycoplasma pneumoniae* cells treated with chloramphenicol. Similar to EMPIAR-10304, due to the prohibitive difficulty of exhaustive manual annotation, 3D ground truth coordinates were generated using an established size-based particle detection pipeline [40, 41] with optimized parameters, followed by meticulous manual examination to ensure high fidelity. While a small fraction of particles may remain unlabeled, these annotations are qualitatively robust for empirical benchmarking (see Fig. A20). Notably, our computational efficiency benchmarks (inference runtime and VRAM consumption) are entirely independent of label completeness.

Simulation (Unseen). We utilized 10 external proteins (i.e., strictly disjoint from the training set) sourced from prior studies [9, 42] to generate 20 instances: `1E9R`, `1F1B`, `1FPY`, `1JZ8`, `1OAO`, `2DF7`, `2H12`, `3HHB`, `4D4R`, and `6T3E`. Following the same generation protocol used for the training data, we controlled the particle count to maintain consistent density: 100 particles for single-class volumes (10 instances, systematically cycling through all 10 proteins), 50 particles per class for two-class volumes (5 instances, randomly selected proteins), and 25 particles per class for four-class volumes (5 instances, randomly selected proteins).

Appendix D.2.2 Non-Zero-Shot Setting

Simulation (Seen). Due to the large computational overhead required for exhaustive benchmarking and our primary architectural focus on the zero-shot setting, we evaluated non-zero-shot performance on a random subset of PDBs from the training set. Specifically, we randomly sampled 10 proteins from the 119-protein training set to generate 20 instances: `1BXN`, `1S3X`, `3CF3`, `3GL1`, `5JH9`, `5OOL`, `6IGC`, `6LXV`, `7DD9`, and `7ETM`. The generation protocol and particle density distributions precisely mirror those of the "unseen" simulation dataset (10 single-class, 5 two-class, and 5 four-class instances).

Appendix E Benchmark Setting Details

Appendix E.1 Prompt Quantities and Evaluation Trials

We evaluated all open-set methods under 1-, 2-, and 4-prompt settings. We capped the maximum number of visual prompts at four for two primary reasons. First, this represents a realistic empirical constraint: because the sparsest target class across our evaluation datasets contains only nine ground truth particles (Table A2), supplying four prompts would already consume nearly half of the available targets in the volume. Second, manually identifying four reference particles requires negligible human effort (taking only seconds for an annotator), strictly adhering to the fundamental objective of open-set detection paradigms to minimize annotation burden.

To ensure statistical robustness, all open-set experiments were averaged across 10 independent trials. For each trial, the reference prompts were randomly re-sampled from the available ground truth pool. In contrast, due to the prohibitive computational cost of repeatedly training models from scratch, evaluations for the WSL baselines were limited to three independent trials.

Appendix E.2 Auxiliary F1 Score Metric

Because several baseline methods do not output confidence scores, calculating the standard mean Average Precision (mAP) across all evaluated frameworks is not feasible. To ensure a fair and comprehensive comparison, we report the F1 score as an auxiliary evaluation metric. For methods that do yield confidence scores, we determine the optimal F1 score by performing a discrete grid search over the confidence threshold space. Specifically, we sweep the threshold from 0.0 to 1.0 at intervals of 0.05, systematically filtering out low-confidence outputs at each step to report the maximum achieved F1 score for that method.

Appendix E.3 Runtime and Memory Benchmarking Protocol

To effectively isolate the model's intrinsic computational efficiency from highly variable I/O bottlenecks (e.g., disk read latency, CPU page faults, and OS-level scheduling), we strictly measure the core inference execution time. Specifically, runtime is recorded using Python's high-resolution `time.perf_counter()` from the precise moment the data is fully loaded into memory until the final 3D coordinates and associated metadata (e.g., confidence scores) are generated.

Peak VRAM consumption is exclusively profiled within this exact execution window using `torch.cuda.max_memory_allocated()`. To guarantee independent and accurate memory statistics across sequential trials, we actively clear the profiler state via `torch.cuda.reset_peak_memory_stats()` prior to each measurement.

Appendix F Training Details

Appendix F.1 Augmentation

After loading the paired tilt-series images and their 3D ground truth coordinates, we apply consistent 3D-aware augmentations across the series, followed by 2D-only augmentations on individual images. The 3D-aware transformations include random resizing (scale factors 0.7 to 1.0), padding (scale factors 1.0 to 1.1), and spatial flipping. To maintain geometric consistency, a single set of transformation parameters is sampled and applied uniformly to both the 2D tilt images and the corresponding 3D ground truth. To accommodate the 48 GB VRAM limit of our NVIDIA RTX 6000 Ada GPUs, padding is discarded if the combined resizing and padding operations would result in an image larger than the original resolution. Subsequently, 2D-only augmentations are applied independently to each image, which include the addition of random Gaussian noise ($\sigma = 0.1$) and a random grayscale rectangular occlusion (up to 150 pixels in size) to simulate physical occlusion and other adverse imaging artifacts.

Next, we randomly sample the set of input tilts and visual prompts. Specifically, we first determine the number of target classes (ranging from 1 to the total number of unique classes present in the instance). For each selected class, we uniformly sample between 1 and 5 visual prompts. Subsequently, the order of the tilt images and their corresponding prompts are randomly shuffled. This prevents the

model from overfitting to a specific sequential order, forcing it to learn the underlying multi-view geometry. Finally, we randomly sample a subset of 21 to 31 tilt images from the full sequence (also to strictly adhere to the aforementioned VRAM constraints).

During each training epoch, we iterate over the entire simulated cryo-ET dataset, followed by iterations over dynamically generated data from our auxiliary geometric primitives module ($\mathcal{G}$) equivalent to half the size of the simulated dataset. We empirically set these augmentation hyperparameters following established practices in standard visual prompting architectures [43, 32]. Due to the extensive computational cost of training, an exhaustive ablation study of these augmentation hyperparameters was not feasible.

Appendix F.2 Implementation and Training Time

FullTilt is implemented in PyTorch 2.5.1 and trained for 200 epochs using the AdamW optimizer [39] with a learning rate of 1×10^{-4}. Distributed training is conducted across four NVIDIA RTX 6000 Ada GPUs with a per-GPU batch size of 1, requiring approximately two weeks to converge. To better understand the computational bottleneck, we conducted empirical profiling of the training loop. We observed that the core network operations execute efficiently: the forward pass takes approximately 400 ms, the backward pass requires roughly 700 ms, and distributed gradient synchronization (e.g., AllReduce) takes less than 200 ms. In stark contrast, the bipartite matching required for the loss calculation dominates the computational overhead, consuming nearly 2 seconds per iteration.

Appendix G Baseline Implementation Details

Appendix G.1 Open-Set Methods for Cryo-ET Data

TomoTwin [9]. We utilized the official codebase and pretrained weights provided by the authors. Inference was conducted using all officially recommended hyperparameters, with the sole exception of increasing the batch size from 256 to 512. This modification accelerated the embedding process by leveraging the larger VRAM capacity of our NVIDIA RTX 6000 Ada GPU.

CryoSAM [10]. We set up the official environment and loaded the designated pretrained weights for SAM [13]. Despite strictly adhering to the official inference protocols and experimenting with various prompts, we were unable to reproduce the results reported in the original paper. We attribute this failure to a severe zero-shot domain gap: SAM, pretrained on natural images, struggled to generalize to the inherently low-contrast, noisy slices of cryo-ET tomograms, consistently yielding empty or nonsensical segmentation masks (Figs. A3 and A4). This irreproducibility has also been documented in recent literature [11].

ProPicker [11]. We utilized the official implementation with the provided pretrained weights (ProPicker and TomoTwin [9]). Inference was performed exactly according to the officially suggested parameter configurations.

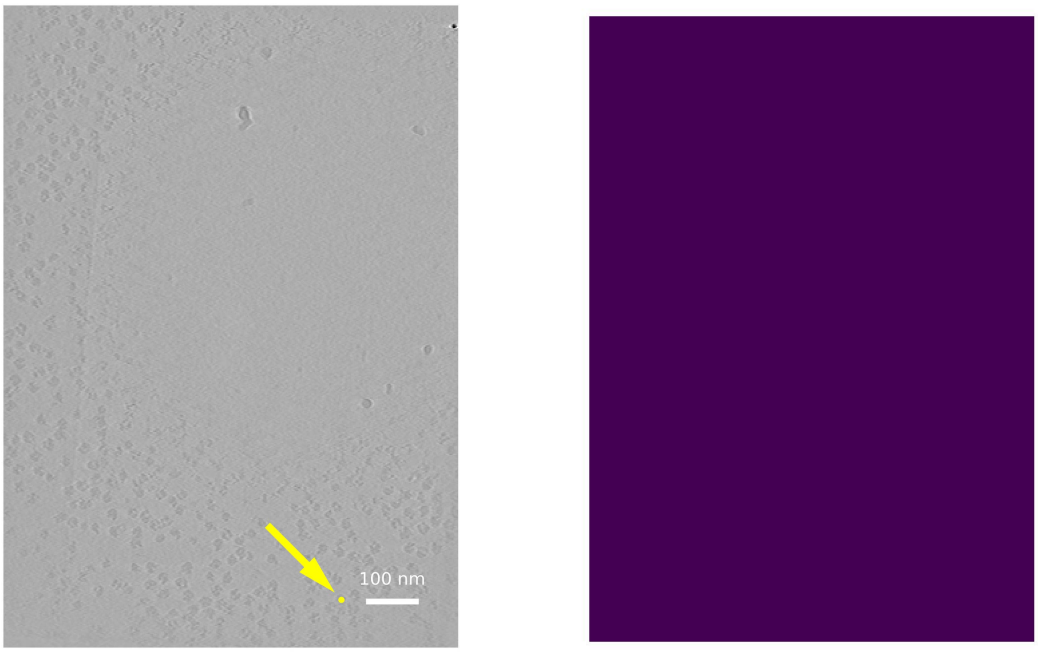
Appendix G.2 Weakly Supervised Learning Methods for Cryo-ET Data

Huang et al. [26]. We configured the environment using the official repository. The model was trained and evaluated strictly following the authors' recommended hyperparameters.

TomoPicker [12]. Using the official codebase, we trained the model with the suggested `GE-KL` objective, which is documented as the optimal configuration. Inference was subsequently conducted using all default hyperparameters.

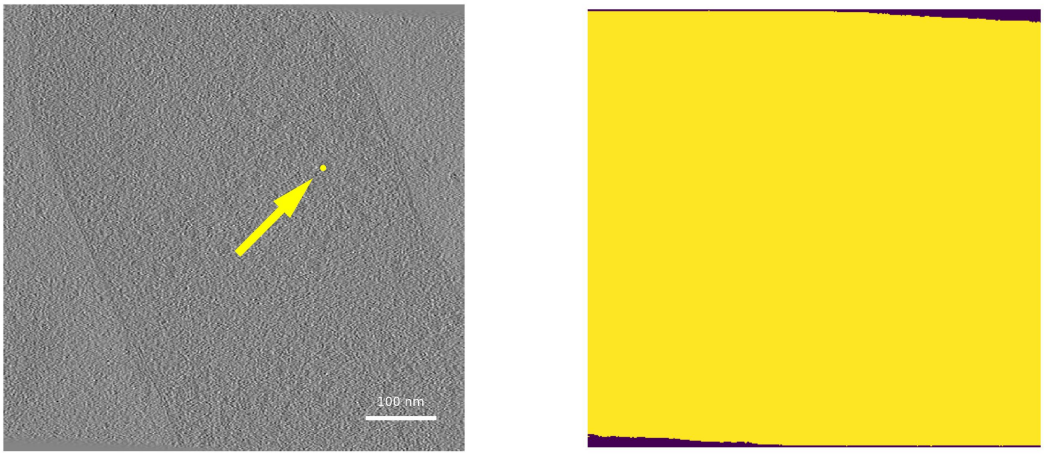
Appendix G.3 Other Methods for Cryo-ET Data

Zeng et al. [22]. As the authors do not provide source code or specific implementation details for their 2D-to-3D backprojection mechanics, we rigorously re-implemented their approach to ensure a fair comparison. While the original paper utilized Faster R-CNN [44] as the 2D detector, we upgraded the architecture to a DETR-based detector to align with our framework's baseline capabilities. Specifically, we integrated DETR with our multiclass visual prompt encoder and replaced the standard decoder with a 3-stage formulation [33]. To handle the inherently high false-positive rate



(a) Prompt (b) Mask

Figure A3: **Example of CryoSAM failure on EMPIAR-10304.** Given the prompt (a small yellow dot indicated by the yellow arrow), CryoSAM fails to estimate any masked region (i.e., yields an empty mask).



(a) Prompt (b) Mask

Figure A4: **Example of CryoSAM failure on EMPIAR-10499.** Given the prompt (a small yellow dot indicated by the yellow arrow), CryoSAM incorrectly estimates a mask covering the entire image.

of independent 2D detections during 3D backprojection, we implemented a RANSAC-based [45] algorithm. The model was trained using the exact same dataset and hyperparameters as FullTilt. Training was conducted using PyTorch 2.5.1 across four NVIDIA RTX 6000 Ada GPUs.

Appendix G.4 Multi-View 3D Object Detection Methods

DETR3D [19]. We adapted the DETR3D architecture by integrating our multiclass visual prompt encoder and swapping its default decoder for a 3-stage decoder [33]. The model was trained using the

exact same dataset as FullTilt, plus data generated by our auxiliary geometric primitives ($\mathcal{G}$) module. Training was executed with PyTorch 2.5.1 on four NVIDIA RTX 6000 Ada GPUs.

PETR [20]. Similar to DETR3D, we integrated our multiclass visual prompt encoder into the PETR architecture and utilized a 3-stage decoder [33]. The model was trained on the identical FullTilt dataset (plus $\mathcal{G}$). Due to the extensively higher VRAM requirements of PETR's global attention mechanism, training was conducted on four NVIDIA H200 GPUs. However, to ensure a fair latency comparison against our method, all inference benchmarking for PETR was performed on a single NVIDIA RTX 6000 Ada GPU.

Appendix H Comprehensive Evaluation

Appendix H.1 Additional Intra-Instance Detection Results

The evaluation results on the two simulation datasets are provided in Tables A3 and A4. While FullTilt demonstrates superior performance on real-world datasets, it slightly underperforms compared to TomoTwin and ProPicker on some metrics within the simulated environments. Nevertheless, FullTilt remains the fastest and most memory-efficient framework among all evaluated open-set cryo-ET detection methods. This performance discrepancy on synthetic data might suggest that TomoTwin and ProPicker (which relies on TomoTwin's pretrained embeddings) overfit to the idealized imaging conditions of the simulation distribution.

Furthermore, we provide detailed class-level performance breakdowns across the DS-10440, "seen"simulation, and "unseen" simulation datasets. These comprehensive metrics are tabulated individually for TomoTwin (Tables A5, A6, and A7), ProPicker (Tables A8, A9, and A10), and FullTilt (Tables A11, A12, and A13).

Table A3: **Intra-instance detection results on the simulation dataset (seen).** The best-performing cryo-ET method for each metric is highlighted in bold.

Category	Method	# Prompts	mAP@0.5r ↑	mAP@1r ↑	F1 ↑	T. ↓	M. ↓
General MV	DETR3D [19]+$\mathcal{E}_{\text{Prompt}}$	1	0.317 ± 0.013	0.520 ± 0.020	0.519 ± 0.007	0.322	4800
	PETR [20]+$\mathcal{E}_{\text{Prompt}}$		0.000 ± 0.000	0.001 ± 0.000	0.009 ± 0.000	0.849	17700
Cryo-ET	Zeng et al. [22]+$\mathcal{E}_{\text{Prompt}}$		n.a.	n.a.	0.003 ± 0.000	16.0	6320
	TomoTwin [9]		0.578 ± 0.018	**0.708 ± 0.021**	**0.732 ± 0.019**	2160	20800
	CryoSAM [10]		n.a.	n.a.	0.003 ± 0.001	296	29100
	ProPicker [11]		n.a.	n.a.	0.683 ± 0.021	65.6	12600
	FullTilt (ours)		**0.632 ± 0.006**	0.670 ± 0.007	0.637 ± 0.008	**0.682**	**5160**
General MV	DETR3D [19]+$\mathcal{E}_{\text{Prompt}}$	2	0.350 ± 0.007	0.541 ± 0.007	0.533 ± 0.003	0.321	4800
	PETR [20]+$\mathcal{E}_{\text{Prompt}}$		0.000 ± 0.000	0.001 ± 0.000	0.009 ± 0.000	0.849	17700
Cryo-ET	Zeng et al. [22]+$\mathcal{E}_{\text{Prompt}}$		n.a.	n.a.	0.002 ± 0.000	16.0	6320
	TomoTwin [9]		0.597 ± 0.014	**0.726 ± 0.020**	**0.757 ± 0.021**	2160	20800
	CryoSAM [10]		n.a.	n.a.	0.003 ± 0.001	295	29100
	ProPicker [11]		n.a.	n.a.	0.696 ± 0.010	64.3	12600
	FullTilt (ours)		**0.631 ± 0.008**	0.670 ± 0.007	0.645 ± 0.009	**0.685**	**5160**
General MV	DETR3D [19]+$\mathcal{E}_{\text{Prompt}}$	4	0.352 ± 0.005	0.533 ± 0.005	0.527 ± 0.002	0.325	4800
	PETR [20]+$\mathcal{E}_{\text{Prompt}}$		0.000 ± 0.000	0.001 ± 0.000	0.009 ± 0.000	0.847	17700
Cryo-ET	Zeng et al. [22]+$\mathcal{E}_{\text{Prompt}}$		n.a.	n.a.	0.002 ± 0.000	16.0	6320
	TomoTwin [9]		0.604 ± 0.013	**0.739 ± 0.026**	**0.772 ± 0.030**	2160	20800
	CryoSAM [10]		n.a.	n.a.	0.004 ± 0.001	297	29200
	ProPicker [11]		n.a.	n.a.	0.700 ± 0.009	64.3	12600
	FullTilt (ours)		**0.627 ± 0.008**	0.662 ± 0.007	0.648 ± 0.005	**0.687**	**5160**

+$\mathcal{E}_{\text{Prompt}}$: with our multiclass visual prompt encoder module; MV: multi-view; n.a.: not applicable because the metric cannot be measured; T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

Table A4: **Zero-shot intra-instance detection results on the simulation dataset (unseen).** The best-performing cryo-ET method for each metric is highlighted in bold.

Category	Method	# Prompts	mAP@0.5r ↑	mAP@1r ↑	F1 ↑	T. ↓	M. ↓
General MV	DETR3D [19]+$\mathcal{E}_{\text{Prompt}}$	1	0.208 ± 0.009	0.392 ± 0.009	0.426 ± 0.006	0.321	4800
	PETR [20]+$\mathcal{E}_{\text{Prompt}}$		0.000 ± 0.000	0.000 ± 0.000	0.004 ± 0.000	0.848	17700
Cryo-ET	Zeng et al. [22]+$\mathcal{E}_{\text{Prompt}}$		n.a.	n.a.	0.001 ± 0.000	17.5	6310
	TomoTwin [9]		**0.531 ± 0.010**	**0.678 ± 0.021**	**0.697 ± 0.025**	2160	20800
	CryoSAM [10]		n.a.	n.a.	0.002 ± 0.001	293	29100
	ProPicker [11]		n.a.	n.a.	0.663 ± 0.017	62.9	12600
	FullTilt (ours)		0.502 ± 0.012	0.542 ± 0.013	0.521 ± 0.012	**0.694**	**5160**
General MV	DETR3D [19]+$\mathcal{E}_{\text{Prompt}}$	2	0.248 ± 0.009	0.418 ± 0.008	0.435 ± 0.003	0.322	4800
	PETR [20]+$\mathcal{E}_{\text{Prompt}}$		0.000 ± 0.000	0.000 ± 0.000	0.004 ± 0.000	0.849	17700
Cryo-ET	Zeng et al. [22]+$\mathcal{E}_{\text{Prompt}}$		n.a.	n.a.	0.001 ± 0.000	17.5	6310
	TomoTwin [9]		**0.549 ± 0.011**	**0.697 ± 0.017**	**0.700 ± 0.019**	2160	20800
	CryoSAM [10]		n.a.	n.a.	0.003 ± 0.001	294	29100
	ProPicker [11]		n.a.	n.a.	0.683 ± 0.013	66.1	12600
	FullTilt (ours)		0.513 ± 0.010	0.552 ± 0.011	0.536 ± 0.015	**0.698**	**5160**
General MV	DETR3D [19]+$\mathcal{E}_{\text{Prompt}}$	4	0.256 ± 0.006	0.415 ± 0.005	0.436 ± 0.003	0.320	4800
	PETR [20]+$\mathcal{E}_{\text{Prompt}}$		0.000 ± 0.000	0.000 ± 0.000	0.004 ± 0.000	0.848	17700
Cryo-ET	Zeng et al. [22]+$\mathcal{E}_{\text{Prompt}}$		n.a.	n.a.	0.001 ± 0.000	17.5	6310
	TomoTwin [9]		**0.563 ± 0.007**	**0.706 ± 0.012**	**0.717 ± 0.015**	2160	20800
	CryoSAM [10]		n.a.	n.a.	0.002 ± 0.000	296	29100
	ProPicker [11]		n.a.	n.a.	0.689 ± 0.011	61.5	12600
	FullTilt (ours)		0.513 ± 0.004	0.553 ± 0.005	0.545 ± 0.005	**0.699**	**5160**

$+\mathcal{E}_{\text{Prompt}}$: with our multiclass visual prompt encoder module; MV: multi-view; n.a.: not applicable because the metric cannot be measured; T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

Table A5: **Detailed zero-shot intra-instance detection results on the DS-10440 dataset (TomoTwin).**

	1 Prompt			2 Prompts			4 Prompts		
Class	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**
apoferritin	0.039 ± 0.030	0.073 ± 0.043	0.102 ± 0.051	0.047 ± 0.021	0.082 ± 0.031	0.146 ± 0.046	0.093 ± 0.057	0.158 ± 0.081	0.210 ± 0.088
β-amylase	0.040 ± 0.012	0.052 ± 0.018	0.029 ± 0.022	0.021 ± 0.014	0.025 ± 0.016	0.032 ± 0.033	0.025 ± 0.010	0.030 ± 0.010	0.043 ± 0.028
β-galactosidase	0.065 ± 0.016	0.095 ± 0.027	0.089 ± 0.023	0.047 ± 0.025	0.069 ± 0.038	0.099 ± 0.057	0.042 ± 0.022	0.063 ± 0.027	0.087 ± 0.043
ribosome	0.079 ± 0.011	0.211 ± 0.023	0.079 ± 0.018	0.061 ± 0.014	0.159 ± 0.044	0.079 ± 0.026	0.063 ± 0.016	0.177 ± 0.044	0.081 ± 0.012
thyroglobulin	0.089 ± 0.012	0.117 ± 0.012	0.121 ± 0.014	0.058 ± 0.011	0.078 ± 0.013	0.103 ± 0.043	0.074 ± 0.016	0.091 ± 0.018	0.118 ± 0.034
virus-like particles	0.222 ± 0.102	0.241 ± 0.108	0.050 ± 0.009	0.165 ± 0.068	0.178 ± 0.069	0.077 ± 0.019	0.107 ± 0.052	0.113 ± 0.055	0.072 ± 0.029
mean	0.089 ± 0.021	0.132 ± 0.024	0.078 ± 0.012	0.066 ± 0.012	0.098 ± 0.015	0.089 ± 0.010	0.067 ± 0.010	0.105 ± 0.015	0.102 ± 0.021

↓: lower is better; ↑: higher is better

Table A6: **Detailed intra-instance detection results on the simulation dataset (seen; TomoTwin).**

	1 Prompt			2 Prompts			4 Prompts		
Class	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**
1BXN	0.812 ± 0.044	0.898 ± 0.035	0.897 ± 0.028	0.854 ± 0.041	0.922 ± 0.030	0.933 ± 0.020	0.878 ± 0.013	0.926 ± 0.015	0.920 ± 0.034
1S3X	0.010 ± 0.009	0.170 ± 0.072	0.234 ± 0.106	0.012 ± 0.009	0.141 ± 0.084	0.192 ± 0.120	0.018 ± 0.014	0.228 ± 0.128	0.291 ± 0.169
3CF3	0.321 ± 0.159	0.527 ± 0.255	0.544 ± 0.236	0.484 ± 0.085	0.664 ± 0.108	0.733 ± 0.087	0.505 ± 0.108	0.687 ± 0.134	0.772 ± 0.109
3GL1	0.010 ± 0.007	0.101 ± 0.050	0.170 ± 0.059	0.013 ± 0.012	0.137 ± 0.073	0.207 ± 0.090	0.018 ± 0.011	0.159 ± 0.080	0.183 ± 0.112
5JH9	0.890 ± 0.012	0.900 ± 0.012	0.911 ± 0.008	0.892 ± 0.004	0.901 ± 0.004	0.926 ± 0.017	0.893 ± 0.002	0.902 ± 0.002	0.934 ± 0.002
5OOL	0.900 ± 0.010	0.903 ± 0.006	0.927 ± 0.019	0.904 ± 0.008	0.905 ± 0.009	0.946 ± 0.005	0.901 ± 0.004	0.901 ± 0.004	0.945 ± 0.003
6IGC	0.912 ± 0.005	0.944 ± 0.006	0.971 ± 0.003	0.910 ± 0.007	0.943 ± 0.005	0.970 ± 0.004	0.914 ± 0.005	0.944 ± 0.007	0.971 ± 0.004
6LXV	0.926 ± 0.005	0.926 ± 0.005	0.958 ± 0.003	0.916 ± 0.016	0.916 ± 0.016	0.950 ± 0.014	0.922 ± 0.007	0.922 ± 0.007	0.956 ± 0.004
7DD9	0.842 ± 0.005	0.842 ± 0.005	0.881 ± 0.005	0.829 ± 0.033	0.833 ± 0.025	0.853 ± 0.007	0.833 ± 0.014	0.833 ± 0.014	0.866 ± 0.005
7ETM	0.151 ± 0.033	0.864 ± 0.053	0.829 ± 0.013	0.156 ± 0.024	0.897 ± 0.029	0.864 ± 0.018	0.158 ± 0.031	0.888 ± 0.036	0.877 ± 0.017
mean	0.578 ± 0.018	0.708 ± 0.021	0.732 ± 0.019	0.597 ± 0.014	0.726 ± 0.020	0.757 ± 0.021	0.604 ± 0.013	0.739 ± 0.026	0.772 ± 0.030

↓: lower is better; ↑: higher is better

Table A7: **Detailed zero-shot intra-instance detection results on the simulation dataset (unseen; TomoTwin).**

	1 Prompt			2 Prompts			4 Prompts		
Class	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**
1E9R	0.885 ± 0.006	0.888 ± 0.007	0.928 ± 0.004	0.880 ± 0.016	0.891 ± 0.012	0.900 ± 0.005	0.883 ± 0.011	0.888 ± 0.009	0.930 ± 0.007
1F1B	0.214 ± 0.053	0.664 ± 0.085	0.670 ± 0.073	0.278 ± 0.050	0.794 ± 0.057	0.782 ± 0.051	0.357 ± 0.050	0.794 ± 0.068	0.801 ± 0.046
1FPY	0.896 ± 0.046	0.929 ± 0.032	0.940 ± 0.031	0.924 ± 0.013	0.935 ± 0.011	0.947 ± 0.016	0.935 ± 0.013	0.942 ± 0.010	0.952 ± 0.009
1JZ8	0.778 ± 0.059	0.830 ± 0.044	0.836 ± 0.044	0.833 ± 0.023	0.844 ± 0.026	0.851 ± 0.014	0.850 ± 0.024	0.864 ± 0.023	0.865 ± 0.022
1OAO	0.691 ± 0.091	0.803 ± 0.071	0.816 ± 0.058	0.723 ± 0.037	0.822 ± 0.046	0.806 ± 0.073	0.718 ± 0.025	0.813 ± 0.036	0.795 ± 0.061
2DF7	0.961 ± 0.007	0.961 ± 0.007	0.978 ± 0.003	0.963 ± 0.005	0.963 ± 0.005	0.981 ± 0.003	0.960 ± 0.005	0.960 ± 0.005	0.980 ± 0.003
2H12	0.728 ± 0.071	0.903 ± 0.030	0.893 ± 0.036	0.706 ± 0.079	0.873 ± 0.085	0.883 ± 0.079	0.750 ± 0.035	0.905 ± 0.015	0.890 ± 0.007
3HHB	0.076 ± 0.029	0.531 ± 0.124	0.589 ± 0.112	0.088 ± 0.037	0.542 ± 0.118	0.528 ± 0.211	0.077 ± 0.019	0.581 ± 0.065	0.619 ± 0.056
4D4R	0.003 ± 0.003	0.023 ± 0.022	0.051 ± 0.032	0.005 ± 0.006	0.026 ± 0.028	0.050 ± 0.050	0.001 ± 0.001	0.012 ± 0.003	0.034 ± 0.018
6T3E	0.079 ± 0.045	0.250 ± 0.085	0.273 ± 0.107	0.093 ± 0.034	0.276 ± 0.097	0.270 ± 0.105	0.100 ± 0.028	0.304 ± 0.081	0.305 ± 0.083
mean	0.531 ± 0.010	0.678 ± 0.021	0.697 ± 0.025	0.549 ± 0.011	0.697 ± 0.017	0.700 ± 0.019	0.563 ± 0.007	0.706 ± 0.012	0.717 ± 0.015

↓: lower is better; ↑: higher is better

Table A8: **Detailed zero-shot intra-instance detection results on the DS-10440 dataset (ProPicker).**

	1 Prompt			2 Prompts			4 Prompts		
Class	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**
apoferritin	n.a.	n.a.	0.131 ± 0.032	n.a.	n.a.	0.159 ± 0.019	n.a.	n.a.	0.175 ± 0.024
β-amylase	n.a.	n.a.	0.013 ± 0.004	n.a.	n.a.	0.016 ± 0.004	n.a.	n.a.	0.018 ± 0.001
β-galactosidase	n.a.	n.a.	0.029 ± 0.004	n.a.	n.a.	0.027 ± 0.003	n.a.	n.a.	0.030 ± 0.003
ribosome	n.a.	n.a.	0.085 ± 0.026	n.a.	n.a.	0.119 ± 0.017	n.a.	n.a.	0.145 ± 0.018
thyroglobulin	n.a.	n.a.	0.071 ± 0.006	n.a.	n.a.	0.073 ± 0.005	n.a.	n.a.	0.077 ± 0.003
virus-like particles	n.a.	n.a.	0.039 ± 0.008	n.a.	n.a.	0.037 ± 0.008	n.a.	n.a.	0.041 ± 0.006
mean	n.a.	n.a.	0.061 ± 0.008	n.a.	n.a.	0.072 ± 0.004	n.a.	n.a.	0.081 ± 0.004

n.a.: not applicable because the metric cannot be measured; ↓: lower is better; ↑: higher is better

Table A9: **Detailed intra-instance detection results on the simulation dataset (seen; ProPicker).**

	1 Prompt			2 Prompts			4 Prompts		
Class	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**
1BXN	n.a.	n.a.	0.814 ± 0.018	n.a.	n.a.	0.803 ± 0.019	n.a.	n.a.	0.810 ± 0.011
1S3X	n.a.	n.a.	0.247 ± 0.106	n.a.	n.a.	0.202 ± 0.047	n.a.	n.a.	0.248 ± 0.064
3CF3	n.a.	n.a.	0.535 ± 0.156	n.a.	n.a.	0.606 ± 0.109	n.a.	n.a.	0.611 ± 0.071
3GL1	n.a.	n.a.	0.153 ± 0.031	n.a.	n.a.	0.189 ± 0.053	n.a.	n.a.	0.166 ± 0.023
5JH9	n.a.	n.a.	0.828 ± 0.001	n.a.	n.a.	0.827 ± 0.001	n.a.	n.a.	0.829 ± 0.002
5OOL	n.a.	n.a.	0.887 ± 0.073	n.a.	n.a.	0.906 ± 0.026	n.a.	n.a.	0.919 ± 0.007
6IGC	n.a.	n.a.	0.882 ± 0.002	n.a.	n.a.	0.882 ± 0.003	n.a.	n.a.	0.878 ± 0.004
6LXV	n.a.	n.a.	0.949 ± 0.004	n.a.	n.a.	0.949 ± 0.004	n.a.	n.a.	0.951 ± 0.008
7DD9	n.a.	n.a.	0.790 ± 0.163	n.a.	n.a.	0.842 ± 0.009	n.a.	n.a.	0.844 ± 0.006
7ETM	n.a.	n.a.	0.745 ± 0.018	n.a.	n.a.	0.749 ± 0.010	n.a.	n.a.	0.743 ± 0.006
mean	n.a.	n.a.	0.683 ± 0.021	n.a.	n.a.	0.696 ± 0.010	n.a.	n.a.	0.700 ± 0.009

n.a.: not applicable because the metric cannot be measured; ↓: lower is better; ↑: higher is better

Table A10: **Detailed zero-shot intra-instance detection results on the simulation dataset (unseen; ProPicker).**

	1 Prompt			2 Prompts			4 Prompts		
Class	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**
1E9R	n.a.	n.a.	0.903 ± 0.004	n.a.	n.a.	0.895 ± 0.013	n.a.	n.a.	0.895 ± 0.006
1F1B	n.a.	n.a.	0.781 ± 0.066	n.a.	n.a.	0.793 ± 0.036	n.a.	n.a.	0.816 ± 0.005
1FPY	n.a.	n.a.	0.756 ± 0.110	n.a.	n.a.	0.812 ± 0.070	n.a.	n.a.	0.826 ± 0.056
1JZ8	n.a.	n.a.	0.762 ± 0.104	n.a.	n.a.	0.783 ± 0.072	n.a.	n.a.	0.786 ± 0.080
1OAO	n.a.	n.a.	0.791 ± 0.096	n.a.	n.a.	0.842 ± 0.028	n.a.	n.a.	0.869 ± 0.022
2DF7	n.a.	n.a.	0.897 ± 0.024	n.a.	n.a.	0.893 ± 0.038	n.a.	n.a.	0.873 ± 0.029
2H12	n.a.	n.a.	0.797 ± 0.035	n.a.	n.a.	0.791 ± 0.017	n.a.	n.a.	0.789 ± 0.015
3HHB	n.a.	n.a.	0.558 ± 0.112	n.a.	n.a.	0.595 ± 0.074	n.a.	n.a.	0.616 ± 0.032
4D4R	n.a.	n.a.	0.105 ± 0.013	n.a.	n.a.	0.120 ± 0.020	n.a.	n.a.	0.117 ± 0.011
6T3E	n.a.	n.a.	0.284 ± 0.076	n.a.	n.a.	0.309 ± 0.059	n.a.	n.a.	0.300 ± 0.043
mean	n.a.	n.a.	0.663 ± 0.017	n.a.	n.a.	0.683 ± 0.013	n.a.	n.a.	0.689 ± 0.011

n.a.: not applicable because the metric cannot be measured; ↓: lower is better; ↑: higher is better

Table A11: **Detailed zero-shot intra-instance detection results on the DS-10440 dataset (FullTilt).**

	1 Prompt			2 Prompts			4 Prompts		
Class	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**
apoferritin	0.002 ± 0.001	0.009 ± 0.002	0.056 ± 0.006	0.002 ± 0.000	0.009 ± 0.001	0.055 ± 0.007	0.003 ± 0.000	0.011 ± 0.001	0.060 ± 0.005
β-amylase	0.018 ± 0.007	0.027 ± 0.010	0.083 ± 0.016	0.021 ± 0.007	0.032 ± 0.010	0.087 ± 0.012	0.024 ± 0.004	0.036 ± 0.005	0.084 ± 0.015
β-galactosidase	0.052 ± 0.015	0.073 ± 0.017	0.132 ± 0.009	0.057 ± 0.009	0.073 ± 0.009	0.144 ± 0.016	0.057 ± 0.008	0.072 ± 0.009	0.142 ± 0.010
ribosome	0.503 ± 0.067	0.581 ± 0.071	0.559 ± 0.048	0.559 ± 0.045	0.641 ± 0.044	0.622 ± 0.046	0.570 ± 0.029	0.654 ± 0.025	0.641 ± 0.030
thyroglobulin	0.087 ± 0.016	0.115 ± 0.018	0.197 ± 0.019	0.087 ± 0.010	0.116 ± 0.009	0.204 ± 0.023	0.101 ± 0.015	0.134 ± 0.014	0.235 ± 0.021
virus-like particles	0.045 ± 0.020	0.084 ± 0.025	0.131 ± 0.038	0.048 ± 0.020	0.094 ± 0.030	0.141 ± 0.035	0.044 ± 0.013	0.098 ± 0.015	0.166 ± 0.018
mean	0.118 ± 0.013	0.148 ± 0.013	0.193 ± 0.008	0.129 ± 0.010	0.161 ± 0.011	0.209 ± 0.013	0.133 ± 0.007	0.180 ± 0.007	0.221 ± 0.008

↓: lower is better; ↑: higher is better

Table A12: **Detailed intra-instance detection results on the simulation dataset (seen; FullTilt).**

	1 Prompt			2 Prompts			4 Prompts		
Class	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**
1BXN	0.802 ± 0.053	0.833 ± 0.054	0.770 ± 0.062	0.847 ± 0.017	0.876 ± 0.018	0.809 ± 0.028	0.844 ± 0.007	0.870 ± 0.007	0.818 ± 0.009
1S3X	0.000 ± 0.000	0.000 ± 0.000	0.003 ± 0.002	0.000 ± 0.000	0.000 ± 0.000	0.002 ± 0.003	0.000 ± 0.000	0.000 ± 0.000	0.001 ± 0.001
3CF3	0.583 ± 0.015	0.647 ± 0.015	0.588 ± 0.017	0.575 ± 0.016	0.645 ± 0.015	0.586 ± 0.013	0.566 ± 0.012	0.634 ± 0.013	0.576 ± 0.011
3GL1	0.000 ± 0.000	0.000 ± 0.000	0.003 ± 0.002	0.000 ± 0.000	0.000 ± 0.000	0.004 ± 0.002	0.000 ± 0.000	0.000 ± 0.000	0.003 ± 0.002
5JH9	0.786 ± 0.056	0.808 ± 0.055	0.768 ± 0.019	0.779 ± 0.029	0.809 ± 0.030	0.775 ± 0.022	0.783 ± 0.045	0.808 ± 0.042	0.791 ± 0.003
5OOL	0.806 ± 0.035	0.839 ± 0.035	0.828 ± 0.007	0.790 ± 0.017	0.824 ± 0.017	0.840 ± 0.008	0.783 ± 0.006	0.814 ± 0.008	0.843 ± 0.005
6IGC	0.985 ± 0.004	0.991 ± 0.002	0.987 ± 0.003	0.984 ± 0.002	0.992 ± 0.002	0.987 ± 0.002	0.984 ± 0.003	0.991 ± 0.003	0.986 ± 0.003
6LXV	0.949 ± 0.018	0.951 ± 0.017	0.862 ± 0.024	0.957 ± 0.008	0.959 ± 0.006	0.870 ± 0.021	0.961 ± 0.004	0.962 ± 0.004	0.891 ± 0.009
7DD9	0.809 ± 0.065	0.827 ± 0.068	0.809 ± 0.013	0.779 ± 0.070	0.792 ± 0.066	0.806 ± 0.010	0.725 ± 0.034	0.738 ± 0.034	0.802 ± 0.008
7ETM	0.602 ± 0.027	0.799 ± 0.020	0.758 ± 0.048	0.601 ± 0.026	0.800 ± 0.017	0.770 ± 0.047	0.625 ± 0.017	0.805 ± 0.016	0.770 ± 0.038
mean	0.632 ± 0.006	0.670 ± 0.007	0.637 ± 0.008	0.631 ± 0.008	0.670 ± 0.007	0.645 ± 0.009	0.627 ± 0.008	0.662 ± 0.007	0.648 ± 0.005

↓: lower is better; ↑: higher is better

Table A13: **Detailed zero-shot intra-instance detection results on the simulation dataset (unseen; FullTilt).**

	1 Prompt			2 Prompts			4 Prompts		
Class	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**
1E9R	0.654 ± 0.055	0.711 ± 0.051	0.681 ± 0.013	0.662 ± 0.047	0.720 ± 0.045	0.683 ± 0.013	0.661 ± 0.015	0.715 ± 0.020	0.666 ± 0.031
1F1B	0.281 ± 0.044	0.486 ± 0.060	0.596 ± 0.030	0.278 ± 0.025	0.485 ± 0.040	0.612 ± 0.016	0.281 ± 0.008	0.491 ± 0.016	0.618 ± 0.010
1FPY	0.860 ± 0.039	0.864 ± 0.038	0.789 ± 0.011	0.895 ± 0.013	0.899 ± 0.013	0.805 ± 0.006	0.910 ± 0.008	0.915 ± 0.007	0.819 ± 0.007
1JZ8	0.850 ± 0.064	0.857 ± 0.065	0.774 ± 0.014	0.893 ± 0.020	0.898 ± 0.021	0.813 ± 0.037	0.902 ± 0.009	0.908 ± 0.008	0.831 ± 0.004
1OAO	0.612 ± 0.044	0.660 ± 0.046	0.685 ± 0.016	0.618 ± 0.044	0.664 ± 0.048	0.674 ± 0.012	0.586 ± 0.039	0.631 ± 0.042	0.680 ± 0.012
2DF7	0.927 ± 0.042	0.968 ± 0.042	0.891 ± 0.123	0.941 ± 0.070	0.975 ± 0.070	0.944 ± 0.150	0.965 ± 0.003	0.999 ± 0.001	0.996 ± 0.001
2H12	0.832 ± 0.015	0.863 ± 0.013	0.766 ± 0.055	0.840 ± 0.008	0.874 ± 0.008	0.805 ± 0.022	0.826 ± 0.010	0.866 ± 0.011	0.808 ± 0.017
3HHB	0.000 ± 0.000	0.003 ± 0.001	0.014 ± 0.005	0.000 ± 0.000	0.003 ± 0.001	0.014 ± 0.007	0.000 ± 0.000	0.001 ± 0.000	0.009 ± 0.004
4D4R	0.000 ± 0.000	0.000 ± 0.000	0.001 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.001 ± 0.001	0.000 ± 0.000	0.000 ± 0.000	0.001 ± 0.001
6T3E	0.000 ± 0.001	0.005 ± 0.003	0.016 ± 0.008	0.000 ± 0.000	0.002 ± 0.003	0.013 ± 0.010	0.000 ± 0.000	0.002 ± 0.001	0.019 ± 0.008
mean	0.502 ± 0.012	0.542 ± 0.013	0.521 ± 0.012	0.513 ± 0.010	0.552 ± 0.011	0.536 ± 0.015	0.513 ± 0.004	0.553 ± 0.005	0.545 ± 0.005

↓: lower is better; ↑: higher is better

Appendix H.2 Additional Cross-Instance Detection Results

We present cross-instance detection results on the simulated cryo-ET datasets in Table A14, evaluating both the non-zero-shot setting (seen) and the zero-shot setting (unseen). FullTilt achieves the highest mAP@0.5r on the "seen" dataset, while consistently securing the second-best performance across the majority of the remaining metrics and dataset splits.

Furthermore, we provide detailed class-level performance breakdowns across the DS-10440, "seen" simulation, and "unseen" simulation datasets. These comprehensive metrics are tabulated individually for TomoTwin (Tables A15, A16, and A17), ProPicker (Tables A18, A19, and A20), and FullTilt (Tables A21, A22, and A23).

Table A14: **Cross-instance detection results on simulated cryo-ET datasets.** The best-performing open-set method for each metric is highlighted in bold.

			Simulation (Seen)			Simulation (Unseen)		
Cat.	**Method**	**# P.**	**mAP@0.5r ↑**	**mAP@1r ↑**	**F1 ↑**	**mAP@0.5r ↑**	**mAP@1r ↑**	**F1 ↑**
WSL	Huang et al. [26]	1	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
	TomoPicker [12]		0.028 ± 0.018	0.108 ± 0.037	0.138 ± 0.054	0.010 ± 0.003	0.072 ± 0.010	0.088 ± 0.010
OS	TomoTwin [9]		0.582 ± 0.023	**0.730 ± 0.033**	**0.761 ± 0.036**	**0.543 ± 0.029**	**0.677 ± 0.049**	**0.695 ± 0.050**
	ProPicker [11]		n.a.	n.a.	0.707 ± 0.029	n.a.	n.a.	0.674 ± 0.063
	FullTilt (ours)		**0.633 ± 0.006**	0.672 ± 0.007	0.632 ± 0.007	0.500 ± 0.022	0.538 ± 0.021	0.514 ± 0.028
WSL	Huang et al. [26]	2	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
	TomoPicker [12]		0.062 ± 0.006	0.167 ± 0.013	0.189 ± 0.029	0.019 ± 0.008	0.128 ± 0.013	0.125 ± 0.028
OS	TomoTwin [9]		0.584 ± 0.034	**0.749 ± 0.027**	**0.781 ± 0.027**	**0.564 ± 0.022**	**0.728 ± 0.025**	**0.710 ± 0.039**
	ProPicker [11]		n.a.	n.a.	0.714 ± 0.035	n.a.	n.a.	0.708 ± 0.022
	FullTilt (ours)		**0.636 ± 0.008**	0.671 ± 0.007	0.636 ± 0.008	0.510 ± 0.005	0.546 ± 0.005	0.528 ± 0.008
WSL	Huang et al. [26]	4	0.000 ± 0.000	0.001 ± 0.000	0.003 ± 0.003	0.004 ± 0.004	0.012 ± 0.014	0.015 ± 0.017
	TomoPicker [12]		0.068 ± 0.018	0.209 ± 0.023	0.197 ± 0.028	0.029 ± 0.002	0.132 ± 0.032	0.110 ± 0.002
OS	TomoTwin [9]		0.608 ± 0.010	**0.760 ± 0.017**	**0.773 ± 0.023**	**0.573 ± 0.017**	**0.741 ± 0.016**	**0.715 ± 0.029**
	ProPicker [11]		n.a.	n.a.	0.714 ± 0.015	n.a.	n.a.	0.694 ± 0.027
	FullTilt (ours)		**0.625 ± 0.006**	0.660 ± 0.006	0.635 ± 0.006	0.508 ± 0.005	0.545 ± 0.005	0.532 ± 0.008

Cat.: category; # P.: # prompts; WSL: weakly-supervised learning; OS: open-set; -: Huang et al. needs at least 2 prompts; n.a.: not applicable because the metric cannot be measured; ↓: lower is better; ↑: higher is better

Table A15: **Detailed zero-shot cross-instance detection results on the DS-10440 dataset (TomoTwin).**

	1 Prompt			2 Prompts			4 Prompts		
Class	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**
apoferritin	0.037 ± 0.032	0.079 ± 0.072	0.125 ± 0.135	0.116 ± 0.070	0.202 ± 0.108	0.161 ± 0.147	0.123 ± 0.072	0.214 ± 0.111	0.191 ± 0.149
β-amylase	0.024 ± 0.016	0.030 ± 0.022	0.047 ± 0.048	0.003 ± 0.003	0.006 ± 0.005	0.010 ± 0.011	0.003 ± 0.003	0.007 ± 0.005	0.017 ± 0.022
β-galactosidase	0.051 ± 0.034	0.086 ± 0.045	0.108 ± 0.055	0.027 ± 0.016	0.047 ± 0.020	0.048 ± 0.043	0.033 ± 0.009	0.058 ± 0.015	0.053 ± 0.044
ribosome	0.085 ± 0.019	0.235 ± 0.043	0.089 ± 0.079	0.081 ± 0.016	0.233 ± 0.035	0.079 ± 0.054	0.087 ± 0.016	0.244 ± 0.035	0.136 ± 0.123
thyroglobulin	0.072 ± 0.034	0.097 ± 0.038	0.105 ± 0.038	0.095 ± 0.030	0.119 ± 0.030	0.091 ± 0.054	0.090 ± 0.021	0.117 ± 0.020	0.113 ± 0.077
virus-like particles	0.193 ± 0.142	0.200 ± 0.152	0.093 ± 0.070	0.223 ± 0.140	0.234 ± 0.150	0.078 ± 0.094	0.301 ± 0.041	0.321 ± 0.045	0.156 ± 0.138
mean	0.077 ± 0.026	0.121 ± 0.029	0.095 ± 0.026	0.091 ± 0.044	0.140 ± 0.057	0.078 ± 0.042	0.106 ± 0.026	0.160 ± 0.038	0.111 ± 0.061

↓: lower is better; ↑: higher is better

Table A16: **Detailed cross-instance detection results on the simulation dataset (seen; TomoTwin).**

	1 Prompt			2 Prompts			4 Prompts		
Class	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**
1BXN	0.832 ± 0.088	0.929 ± 0.015	0.933 ± 0.022	0.761 ± 0.244	0.914 ± 0.049	0.865 ± 0.049	0.894 ± 0.052	0.930 ± 0.009	0.909 ± 0.026
1S3X	0.019 ± 0.018	0.295 ± 0.132	0.416 ± 0.151	0.034 ± 0.019	0.346 ± 0.119	0.401 ± 0.132	0.020 ± 0.011	0.296 ± 0.141	0.364 ± 0.207
3CF3	0.327 ± 0.211	0.526 ± 0.318	0.545 ± 0.304	0.384 ± 0.177	0.578 ± 0.222	0.678 ± 0.192	0.505 ± 0.085	0.689 ± 0.072	0.763 ± 0.092
3GL1	0.012 ± 0.013	0.130 ± 0.114	0.157 ± 0.135	0.024 ± 0.019	0.237 ± 0.135	0.286 ± 0.162	0.026 ± 0.018	0.262 ± 0.118	0.129 ± 0.127
5JH9	0.897 ± 0.008	0.907 ± 0.008	0.936 ± 0.005	0.895 ± 0.006	0.904 ± 0.006	0.935 ± 0.004	0.892 ± 0.005	0.902 ± 0.005	0.934 ± 0.003
5OOL	0.906 ± 0.006	0.906 ± 0.006	0.947 ± 0.003	0.903 ± 0.007	0.903 ± 0.007	0.944 ± 0.003	0.901 ± 0.005	0.901 ± 0.005	0.945 ± 0.002
6IGC	0.914 ± 0.005	0.947 ± 0.005	0.972 ± 0.003	0.908 ± 0.010	0.939 ± 0.011	0.969 ± 0.006	0.916 ± 0.003	0.946 ± 0.004	0.972 ± 0.002
6LXV	0.911 ± 0.021	0.911 ± 0.021	0.944 ± 0.022	0.922 ± 0.009	0.922 ± 0.009	0.956 ± 0.003	0.926 ± 0.008	0.926 ± 0.008	0.957 ± 0.004
7DD9	0.841 ± 0.005	0.841 ± 0.005	0.882 ± 0.008	0.844 ± 0.007	0.844 ± 0.007	0.885 ± 0.006	0.842 ± 0.006	0.842 ± 0.006	0.875 ± 0.007
7ETM	0.160 ± 0.023	0.906 ± 0.017	0.881 ± 0.015	0.160 ± 0.028	0.902 ± 0.017	0.886 ± 0.008	0.154 ± 0.018	0.911 ± 0.011	0.878 ± 0.022
mean	0.582 ± 0.023	0.730 ± 0.033	0.761 ± 0.036	0.584 ± 0.034	0.749 ± 0.027	0.781 ± 0.027	0.608 ± 0.010	0.760 ± 0.017	0.773 ± 0.023

↓: lower is better; ↑: higher is better

Table A17: **Detailed zero-shot cross-instance detection results on the simulation dataset (unseen; TomoTwin).**

	1 Prompt			2 Prompts			4 Prompts		
Class	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**
1E9R	0.882 ± 0.007	0.890 ± 0.011	0.928 ± 0.007	0.883 ± 0.007	0.897 ± 0.012	0.936 ± 0.010	0.886 ± 0.007	0.886 ± 0.007	0.930 ± 0.006
1F1B	0.256 ± 0.101	0.716 ± 0.178	0.734 ± 0.177	0.316 ± 0.098	0.814 ± 0.073	0.812 ± 0.063	0.330 ± 0.102	0.858 ± 0.040	0.842 ± 0.045
1FPY	0.884 ± 0.076	0.917 ± 0.057	0.930 ± 0.023	0.928 ± 0.019	0.939 ± 0.023	0.950 ± 0.015	0.924 ± 0.021	0.938 ± 0.015	0.939 ± 0.016
1JZ8	0.830 ± 0.031	0.854 ± 0.033	0.854 ± 0.017	0.846 ± 0.027	0.857 ± 0.029	0.854 ± 0.015	0.847 ± 0.019	0.859 ± 0.016	0.860 ± 0.013
1OAO	0.697 ± 0.234	0.773 ± 0.255	0.775 ± 0.263	0.695 ± 0.151	0.798 ± 0.181	0.736 ± 0.246	0.762 ± 0.090	0.846 ± 0.064	0.747 ± 0.054
2DF7	0.961 ± 0.006	0.961 ± 0.006	0.980 ± 0.003	0.960 ± 0.006	0.960 ± 0.006	0.980 ± 0.003	0.960 ± 0.007	0.960 ± 0.007	0.980 ± 0.003
2H12	0.757 ± 0.041	0.915 ± 0.009	0.917 ± 0.018	0.764 ± 0.055	0.913 ± 0.008	0.909 ± 0.007	0.778 ± 0.038	0.910 ± 0.011	0.882 ± 0.015
3HHB	0.062 ± 0.052	0.426 ± 0.288	0.454 ± 0.290	0.086 ± 0.023	0.627 ± 0.090	0.542 ± 0.278	0.119 ± 0.020	0.688 ± 0.049	0.577 ± 0.124
4D4R	0.006 ± 0.006	0.062 ± 0.067	0.094 ± 0.082	0.006 ± 0.004	0.067 ± 0.049	0.051 ± 0.068	0.007 ± 0.004	0.109 ± 0.062	0.110 ± 0.120
6T3E	0.098 ± 0.081	0.261 ± 0.158	0.282 ± 0.182	0.153 ± 0.061	0.404 ± 0.122	0.331 ± 0.164	0.117 ± 0.067	0.355 ± 0.127	0.280 ± 0.169
mean	0.543 ± 0.029	0.677 ± 0.049	0.695 ± 0.050	0.564 ± 0.022	0.728 ± 0.025	0.710 ± 0.039	0.573 ± 0.017	0.741 ± 0.016	0.715 ± 0.029

↓: lower is better; ↑: higher is better

Table A18: **Detailed zero-shot cross-instance detection results on the DS-10440 dataset (ProPicker).**

	1 Prompt			2 Prompts			4 Prompts		
Class	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**
apoferritin	n.a.	n.a.	0.175 ± 0.067	n.a.	n.a.	0.141 ± 0.062	n.a.	n.a.	0.188 ± 0.041
β-amylase	n.a.	n.a.	0.017 ± 0.011	n.a.	n.a.	0.015 ± 0.010	n.a.	n.a.	0.015 ± 0.007
β-galactosidase	n.a.	n.a.	0.028 ± 0.013	n.a.	n.a.	0.026 ± 0.008	n.a.	n.a.	0.035 ± 0.011
ribosome	n.a.	n.a.	0.119 ± 0.042	n.a.	n.a.	0.118 ± 0.067	n.a.	n.a.	0.141 ± 0.044
thyroglobulin	n.a.	n.a.	0.073 ± 0.006	n.a.	n.a.	0.077 ± 0.007	n.a.	n.a.	0.078 ± 0.006
virus-like particles	n.a.	n.a.	0.042 ± 0.011	n.a.	n.a.	0.031 ± 0.020	n.a.	n.a.	0.042 ± 0.019
mean	n.a.	n.a.	0.076 ± 0.013	n.a.	n.a.	0.068 ± 0.018	n.a.	n.a.	0.083 ± 0.011

n.a.: not applicable because the metric cannot be measured; ↓: lower is better; ↑: higher is better

Table A19: **Detailed cross-instance detection results on the simulation dataset (seen; ProPicker).**

	1 Prompt			2 Prompts			4 Prompts		
Class	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**
1BXN	n.a.	n.a.	0.804 ± 0.036	n.a.	n.a.	0.810 ± 0.024	n.a.	n.a.	0.798 ± 0.011
1S3X	n.a.	n.a.	0.285 ± 0.160	n.a.	n.a.	0.355 ± 0.141	n.a.	n.a.	0.269 ± 0.099
3CF3	n.a.	n.a.	0.610 ± 0.145	n.a.	n.a.	0.672 ± 0.106	n.a.	n.a.	0.692 ± 0.063
3GL1	n.a.	n.a.	0.277 ± 0.166	n.a.	n.a.	0.203 ± 0.110	n.a.	n.a.	0.200 ± 0.078
5JH9	n.a.	n.a.	0.830 ± 0.002	n.a.	n.a.	0.828 ± 0.002	n.a.	n.a.	0.826 ± 0.006
5OOL	n.a.	n.a.	0.916 ± 0.013	n.a.	n.a.	0.910 ± 0.006	n.a.	n.a.	0.921 ± 0.009
6IGC	n.a.	n.a.	0.880 ± 0.003	n.a.	n.a.	0.883 ± 0.003	n.a.	n.a.	0.880 ± 0.004
6LXV	n.a.	n.a.	0.948 ± 0.002	n.a.	n.a.	0.948 ± 0.002	n.a.	n.a.	0.951 ± 0.006
7DD9	n.a.	n.a.	0.847 ± 0.005	n.a.	n.a.	0.843 ± 0.008	n.a.	n.a.	0.848 ± 0.005
7ETM	n.a.	n.a.	0.679 ± 0.204	n.a.	n.a.	0.686 ± 0.207	n.a.	n.a.	0.752 ± 0.007
mean	n.a.	n.a.	0.707 ± 0.029	n.a.	n.a.	0.714 ± 0.035	n.a.	n.a.	0.714 ± 0.015

n.a.: not applicable because the metric cannot be measured; ↓: lower is better; ↑: higher is better

Table A20: **Detailed zero-shot cross-instance detection results on the simulation dataset (unseen; ProPicker).**

	1 Prompt			2 Prompts			4 Prompts		
Class	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**
1E9R	n.a.	n.a.	0.901 ± 0.003	n.a.	n.a.	0.901 ± 0.007	n.a.	n.a.	0.900 ± 0.007
1F1B	n.a.	n.a.	0.793 ± 0.075	n.a.	n.a.	0.818 ± 0.005	n.a.	n.a.	0.821 ± 0.002
1FPY	n.a.	n.a.	0.717 ± 0.158	n.a.	n.a.	0.849 ± 0.111	n.a.	n.a.	0.831 ± 0.121
1JZ8	n.a.	n.a.	0.762 ± 0.086	n.a.	n.a.	0.811 ± 0.005	n.a.	n.a.	0.814 ± 0.008
1OAO	n.a.	n.a.	0.729 ± 0.187	n.a.	n.a.	0.833 ± 0.133	n.a.	n.a.	0.898 ± 0.032
2DF7	n.a.	n.a.	0.819 ± 0.225	n.a.	n.a.	0.889 ± 0.045	n.a.	n.a.	0.893 ± 0.030
2H12	n.a.	n.a.	0.792 ± 0.015	n.a.	n.a.	0.793 ± 0.013	n.a.	n.a.	0.796 ± 0.016
3HHB	n.a.	n.a.	0.644 ± 0.055	n.a.	n.a.	0.607 ± 0.099	n.a.	n.a.	0.561 ± 0.128
4D4R	n.a.	n.a.	0.117 ± 0.063	n.a.	n.a.	0.172 ± 0.104	n.a.	n.a.	0.123 ± 0.036
6T3E	n.a.	n.a.	0.466 ± 0.164	n.a.	n.a.	0.411 ± 0.221	n.a.	n.a.	0.302 ± 0.180
mean	n.a.	n.a.	0.674 ± 0.063	n.a.	n.a.	0.708 ± 0.022	n.a.	n.a.	0.694 ± 0.027

n.a.: not applicable because the metric cannot be measured; ↓: lower is better; ↑: higher is better

Table A21: **Detailed zero-shot cross-instance detection results on the DS-10440 dataset (FullTilt).**

	1 Prompt			2 Prompts			4 Prompts		
Class	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**
apoferritin	0.002 ± 0.001	0.009 ± 0.002	0.048 ± 0.015	0.002 ± 0.001	0.009 ± 0.002	0.049 ± 0.014	0.003 ± 0.001	0.010 ± 0.002	0.053 ± 0.010
β-amylase	0.009 ± 0.003	0.014 ± 0.004	0.053 ± 0.020	0.008 ± 0.003	0.014 ± 0.003	0.054 ± 0.024	0.011 ± 0.003	0.016 ± 0.004	0.066 ± 0.018
β-galactosidase	0.033 ± 0.012	0.050 ± 0.016	0.104 ± 0.024	0.041 ± 0.011	0.056 ± 0.013	0.116 ± 0.037	0.041 ± 0.014	0.055 ± 0.017	0.106 ± 0.047
ribosome	0.534 ± 0.191	0.601 ± 0.211	0.544 ± 0.181	0.585 ± 0.104	0.659 ± 0.112	0.584 ± 0.150	0.633 ± 0.049	0.715 ± 0.048	0.607 ± 0.064
thyroglobulin	0.064 ± 0.018	0.086 ± 0.024	0.159 ± 0.039	0.065 ± 0.019	0.086 ± 0.024	0.160 ± 0.036	0.080 ± 0.010	0.106 ± 0.012	0.183 ± 0.037
virus-like particles	0.026 ± 0.023	0.059 ± 0.037	0.096 ± 0.065	0.041 ± 0.027	0.096 ± 0.045	0.155 ± 0.066	0.049 ± 0.014	0.111 ± 0.018	0.183 ± 0.026
mean	0.112 ± 0.033	0.137 ± 0.036	0.167 ± 0.037	0.124 ± 0.020	0.154 ± 0.022	0.186 ± 0.025	0.136 ± 0.009	0.169 ± 0.009	0.200 ± 0.013

↓: lower is better; ↑: higher is better

Table A22: **Detailed cross-instance detection results on the simulation dataset (seen; FullTilt).**

	1 Prompt			2 Prompts			4 Prompts		
Class	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**
1BXN	0.833 ± 0.019	0.865 ± 0.015	0.763 ± 0.010	0.842 ± 0.012	0.868 ± 0.013	0.795 ± 0.026	0.838 ± 0.012	0.864 ± 0.010	0.784 ± 0.018
1S3X	0.000 ± 0.000	0.000 ± 0.000	0.002 ± 0.002	0.000 ± 0.000	0.000 ± 0.000	0.002 ± 0.001	0.000 ± 0.000	0.000 ± 0.000	0.002 ± 0.000
3CF3	0.528 ± 0.037	0.591 ± 0.043	0.559 ± 0.014	0.537 ± 0.031	0.604 ± 0.037	0.568 ± 0.020	0.511 ± 0.045	0.570 ± 0.051	0.554 ± 0.018
3GL1	0.000 ± 0.000	0.000 ± 0.000	0.004 ± 0.002	0.000 ± 0.000	0.000 ± 0.000	0.004 ± 0.002	0.000 ± 0.000	0.000 ± 0.000	0.002 ± 0.001
5JH9	0.843 ± 0.014	0.872 ± 0.015	0.778 ± 0.009	0.825 ± 0.034	0.848 ± 0.038	0.786 ± 0.008	0.799 ± 0.020	0.828 ± 0.020	0.793 ± 0.005
5OOL	0.783 ± 0.024	0.816 ± 0.024	0.828 ± 0.013	0.775 ± 0.009	0.806 ± 0.007	0.837 ± 0.007	0.781 ± 0.007	0.808 ± 0.007	0.839 ± 0.005
6IGC	0.984 ± 0.003	0.991 ± 0.003	0.985 ± 0.003	0.985 ± 0.005	0.992 ± 0.003	0.987 ± 0.003	0.984 ± 0.003	0.991 ± 0.003	0.986 ± 0.004
6LXV	0.943 ± 0.005	0.943 ± 0.005	0.844 ± 0.018	0.951 ± 0.010	0.952 ± 0.009	0.849 ± 0.012	0.950 ± 0.017	0.951 ± 0.017	0.854 ± 0.006
7DD9	0.818 ± 0.043	0.844 ± 0.039	0.809 ± 0.008	0.830 ± 0.033	0.850 ± 0.027	0.804 ± 0.008	0.768 ± 0.042	0.787 ± 0.036	0.799 ± 0.008
7ETM	0.602 ± 0.017	0.795 ± 0.011	0.747 ± 0.044	0.610 ± 0.027	0.793 ± 0.012	0.732 ± 0.052	0.623 ± 0.016	0.802 ± 0.015	0.739 ± 0.054
mean	0.633 ± 0.006	0.672 ± 0.007	0.632 ± 0.007	0.636 ± 0.008	0.671 ± 0.007	0.636 ± 0.008	0.625 ± 0.006	0.660 ± 0.006	0.635 ± 0.006

↓: lower is better; ↑: higher is better

Table A23: **Detailed zero-shot cross-instance detection results on the simulation dataset (unseen; FullTilt).**

	1 Prompt			2 Prompts			4 Prompts		
Class	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**	**AP@0.5r ↑**	**AP@1r ↑**	**F1 ↑**
1E9R	0.652 ± 0.026	0.702 ± 0.021	0.652 ± 0.028	0.640 ± 0.016	0.701 ± 0.014	0.652 ± 0.018	0.644 ± 0.020	0.694 ± 0.020	0.645 ± 0.023
1F1B	0.250 ± 0.018	0.458 ± 0.023	0.589 ± 0.042	0.259 ± 0.009	0.466 ± 0.009	0.609 ± 0.043	0.267 ± 0.007	0.470 ± 0.009	0.616 ± 0.016
1FPY	0.876 ± 0.022	0.880 ± 0.022	0.791 ± 0.014	0.895 ± 0.010	0.899 ± 0.012	0.813 ± 0.008	0.891 ± 0.017	0.898 ± 0.018	0.815 ± 0.016
1JZ8	0.876 ± 0.024	0.881 ± 0.024	0.776 ± 0.048	0.869 ± 0.029	0.874 ± 0.030	0.768 ± 0.034	0.896 ± 0.014	0.903 ± 0.014	0.820 ± 0.016
1OAO	0.625 ± 0.051	0.671 ± 0.052	0.678 ± 0.010	0.620 ± 0.031	0.667 ± 0.034	0.679 ± 0.009	0.588 ± 0.039	0.633 ± 0.045	0.675 ± 0.008
2DF7	0.892 ± 0.204	0.915 ± 0.187	0.823 ± 0.271	0.992 ± 0.018	0.997 ± 0.004	0.979 ± 0.029	0.976 ± 0.035	0.992 ± 0.014	0.959 ± 0.057
2H12	0.831 ± 0.009	0.864 ± 0.008	0.787 ± 0.031	0.825 ± 0.019	0.857 ± 0.014	0.764 ± 0.037	0.819 ± 0.011	0.856 ± 0.012	0.778 ± 0.030
3HHB	0.000 ± 0.000	0.004 ± 0.002	0.017 ± 0.005	0.000 ± 0.000	0.001 ± 0.001	0.003 ± 0.002	0.000 ± 0.000	0.000 ± 0.000	0.004 ± 0.002
4D4R	0.000 ± 0.000	0.000 ± 0.000	0.001 ± 0.001	0.000 ± 0.000	0.000 ± 0.000	0.001 ± 0.001	0.000 ± 0.000	0.000 ± 0.000	0.001 ± 0.001
6T3E	0.001 ± 0.002	0.005 ± 0.003	0.020 ± 0.010	0.000 ± 0.000	0.002 ± 0.002	0.016 ± 0.014	0.000 ± 0.000	0.001 ± 0.001	0.011 ± 0.005
mean	0.500 ± 0.022	0.538 ± 0.021	0.514 ± 0.028	0.510 ± 0.005	0.546 ± 0.005	0.528 ± 0.008	0.508 ± 0.005	0.545 ± 0.005	0.532 ± 0.008

↓: lower is better; ↑: higher is better

Appendix H.3 Additional Multiclass Prompting Detection Results

We evaluate our multiclass visual prompt encoder ($\mathcal{E}_{\text{Prompt}}$) under cross-instance scenarios using the 6-class DS-10440 dataset. To ensure a comprehensive evaluation, we systematically test all possible class combinations (e.g., $\binom{6}{3} = 15$ unique combinations for a 3-class query) averaged over 10 independent trials (Table A24). While simultaneously querying a larger number of classes incurs a minor trade-off in detection performance, FullTilt efficiently processes all requested classes in a single forward pass, introducing negligible runtime and VRAM overhead.

Table A24: **Multi-class prompting performance comparison (cross-instance).** FullTilt maintains stable computational efficiency regardless of the number of target classes.

	1 Prompt				2 Prompts				4 Prompts			
# Classes	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**
1	0.112 ± 0.033	0.137 ± 0.036	0.494	3460	0.124 ± 0.020	0.154 ± 0.022	0.498	3460	0.136 ± 0.009	0.169 ± 0.009	0.498	3460
2	0.111 ± 0.011	0.139 ± 0.013	0.508	3460	0.123 ± 0.006	0.154 ± 0.008	0.499	3460	0.128 ± 0.004	0.158 ± 0.005	0.498	3460
3	0.108 ± 0.005	0.138 ± 0.006	0.500	3470	0.123 ± 0.003	0.154 ± 0.004	0.502	3470	0.124 ± 0.003	0.154 ± 0.004	0.502	3470
4	0.115 ± 0.005	0.146 ± 0.006	0.505	3470	0.122 ± 0.003	0.154 ± 0.003	0.507	3470	0.124 ± 0.003	0.155 ± 0.003	0.506	3480
5	0.109 ± 0.011	0.138 ± 0.012	0.494	3470	0.119 ± 0.005	0.151 ± 0.005	0.497	3470	0.122 ± 0.004	0.153 ± 0.005	0.518	3470
6	0.109 ± 0.022	0.137 ± 0.026	0.481	3470	0.121 ± 0.003	0.152 ± 0.004	0.487	3470	0.117 ± 0.009	0.148 ± 0.011	0.489	3470

T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

Appendix H.4 Extended Hardware and Compilation Benchmarks

To further evaluate the computational flexibility and efficiency of FullTilt, we benchmarked its inference runtime and memory consumption under two additional execution paradigms: (1) utilizing an optimized static computational graph via PyTorch's `torch.compile` on a GPU, and (2) in a strictly CPU-only environment. All evaluations were conducted under the intra-instance, single-prompt configuration, with results detailed in Table A25.

While the initial static graph compilation incurs a one-time overhead of approximately 6 to 7 minutes, it yields a $1.3\times$ to $1.5\times$ acceleration in subsequent inference runtimes across all datasets. Conversely, executing FullTilt purely on a CPU increases inference time by a factor of 35 to 40 relative to the uncompiled GPU baseline. Notably, despite this expected CPU bottleneck, FullTilt still successfully completes inference within a highly practical window of 10 to 30 seconds per tilt-series. This demonstrates the framework's high accessibility for resource-constrained laboratory settings lacking dedicated high-performance accelerators.

Table A25: **Extended computational efficiency benchmarks (compiled GPU and CPU-only execution).** Inference runtime (T.) and peak VRAM consumption (M.) are evaluated under the intra-instance, single-prompt configuration. For the compiled GPU setting, the one-time static graph compilation overhead is also reported in seconds.

	DS-10440			EMPIAR-10304			EMPIAR-10499			Simulation (Seen)			Simulation (Out-Distribution)		
Hardware	**Compile ↓**	**T. ↓**	**M. ↓**	**Compile ↓**	**T. ↓**	**M. ↓**	**Compile ↓**	**T. ↓**	**M. ↓**	**Compile ↓**	**T. ↓**	**M. ↓**	**Compile ↓**	**T. ↓**	**M. ↓**
GPU	n.a.	0.499	3430	n.a.	0.229	2410	n.a.	0.389	3410	n.a.	0.682	5160	n.a.	0.694	5160
GPU (Compiled)	362	0.385	3830	372	0.149	2110	361	0.274	3080	413	0.509	5090	413	0.508	5090
CPU-Only	n.a.	20.1	n.a.	n.a.	9.33	n.a.	n.a.	13.4	n.a.	n.a.	24.6	n.a.	n.a.	24.6	n.a.

Compile: Static graph compilation time (s); T.: Inference runtime (s); M.: VRAM (MB); ↓: lower is better; n.a.: not applicable

Appendix H.5 Statistical Significance

We report statistical comparisons in Tables A26 and A27 for intra-instance and cross-instance detection, respectively. We use the Wilcoxon signed-rank test for paired comparisons between methods evaluated over 10 trials and the Mann-Whitney U test for comparisons with WSL baselines evaluated over 3 trials versus FullTilt's 10 trials. These sample sizes limit statistical power and, for exact tests, p-value resolution. A non-significant result indicates insufficient evidence of a difference at the current sample size, rather than equivalence.

For intra-instance detection, FullTilt significantly outperforms the competing methods across all real-world datasets, except for comparisons with TomoTwin on EMPIAR-10304: mAP@0.5r shows no significant difference with 1 prompt and is significantly lower with 2 and 4 prompts, while F1 shows no significant difference with 2 prompts. For cross-instance detection, FullTilt significantly outperforms all evaluated methods on EMPIAR-10499. On DS-10440, it significantly outperforms

TomoTwin in mAP@0.5r and F1, with no significant difference in mAP@1r. On EMPIAR-10304, no significant differences from TomoTwin are observed across the evaluated metrics.

We use 10 trials because of the computational cost of the baselines. Running 20 additional trials across all datasets would require approximately 30 minutes for FullTilt but more than two months for the baselines, highlighting FullTilt's efficiency. The cross-instance comparisons with TomoTwin on DS-10440 and EMPIAR-10304 should therefore be interpreted in light of the limited statistical power at this sample size.

Table A26: **Statistical comparisons for intra-instance detection.**

		DS-10440			EMPIAR-10304			EMPIAR-10499		
# P.	**vs. Method**	**mAP@0.5r**	**mAP@1r**	**F1**	**mAP@0.5r**	**mAP@1r**	**F1**	**mAP@0.5r**	**mAP@1r**	**F1**
1	DETR3D	▲**	▲**	▲**	▲**	▲**	▲**	▲**	▲**	▲**
	PETR	▲**	▲**	▲**	▲**	▲**	▲**	▲**	▲**	▲**
	Zeng et al.	–	–	▲**	–	–	▲**	–	–	▲**
	TomoTwin	▲**	▲*	▲**	ns	▲**	▲*	▲**	▲**	▲**
	CryoSAM	–	–	▲**	–	–	▲**	–	–	▲**
	ProPicker	–	–	▲**	–	–	▲**	–	–	▲**
2	DETR3D	▲**	▲**	▲**	▲**	▲**	▲**	▲**	▲**	▲**
	PETR	▲**	▲**	▲**	▲**	▲**	▲**	▲**	▲**	▲**
	Zeng et al.	–	–	▲**	–	–	▲**	–	–	▲**
	TomoTwin	▲**	▲**	▲**	▽*	▲**	ns	▲**	▲**	▲**
	CryoSAM	–	–	▲**	–	–	▲**	–	–	▲**
	ProPicker	–	–	▲**	–	–	▲**	–	–	▲**
4	DETR3D	▲**	▲**	▲**	▲**	▲**	▲**	▲**	▲**	▲**
	PETR	▲**	▲**	▲**	▲**	▲**	▲**	▲**	▲**	▲**
	Zeng et al.	–	–	▲**	–	–	▲**	–	–	▲**
	TomoTwin	▲**	▲**	▲**	▽**	▲**	▲*	▲**	▲**	▲**
	CryoSAM	–	–	▲**	–	–	▲**	–	–	▲**
	ProPicker	–	–	▲**	–	–	▲**	–	–	▲**

Each entry compares FullTilt with the indicated method. # P.: number of prompts; ▲/▽: FullTilt performs significantly better/worse; ns: no significant difference; –: not applicable; *: $p < 0.05$; **: $p < 0.01$.

Table A27: **Statistical comparisons for cross-instance detection.**

		DS-10440			EMPIAR-10304			EMPIAR-10499		
# P.	**vs. Method**	**mAP@0.5r**	**mAP@1r**	**F1**	**mAP@0.5r**	**mAP@1r**	**F1**	**mAP@0.5r**	**mAP@1r**	**F1**
1	TomoPicker	▲**	▲**	▲**	▲**	▲**	▲**	▲**	▲**	▲**
	TomoTwin	▲*	ns	▲**	ns	ns	ns	▲**	▲*	▲**
	ProPicker	–	–	▲**	–	–	▲**	–	–	▲**
2	Huang et al.	▲**	▲**	▲**	▲**	▲**	▲**	▲**	▲**	▲**
	TomoPicker	▲**	▲**	▲**	▲**	▲**	▲**	▲**	▲**	▲**
	TomoTwin	▲*	ns	▲**	ns	ns	ns	▲**	▲**	▲*
	ProPicker	–	–	▲**	–	–	▲**	–	–	▲**
4	Huang et al.	▲**	▲**	▲**	▲**	▲**	▲**	▲**	▲**	▲**
	TomoPicker	▲**	▲**	▲**	▲**	▲**	▲**	▲**	▲**	▲**
	TomoTwin	▲*	ns	▲**	ns	ns	ns	▲**	▲**	▲*
	ProPicker	–	–	▲**	–	–	▲**	–	–	▲**

Each entry compares FullTilt with the indicated method. # P.: number of prompts; ▲/▽: FullTilt performs significantly better/worse; ns: no significant difference; –: not applicable; *: $p < 0.05$; **: $p < 0.01$.

Appendix H.6 Shape Generalization and Failure Analysis

We systematically analyze 20 protein types from simulated data, with axis ratios ranging from approximately 1.1 to 2.9 and molecular weights from approximately 43 to 2,366 kDa, to assess (i) performance on non-globular particles with and without $\mathcal{G}$, (ii) whether $\mathcal{G}$ introduces a globular bias, and (iii) failure cases. The axis ratio is defined as $\sqrt{\lambda_{\max}/\lambda_{\min}}$, where $\lambda_{\max}$ and $\lambda_{\min}$ are the largest and smallest PCA eigenvalues of the particle density. Higher axis ratios indicate less globular shapes. For each protein type, we compare performance with and without $\mathcal{G}$ using a Wilcoxon signed-rank test (Tables A28 and A29; rows sorted by axis ratio).

FullTilt detects both globular particles (1E9R, axis ratio 1.15, AP@0.5r $= 0.65$) and non-globular particles (1OAO, axis ratio 2.94, AP@0.5r $= 0.61$). Significant gains from $\mathcal{G}$ occur across the axis-

Table A28: **Effect of $\mathcal{G}$ on seen proteins in simulated data.**

			$\mathcal{G}$ vs. no $\mathcal{G}$		FullTilt		Without $\mathcal{G}$	
PDB	**Weight (kDa)**	**Axis ratio**	**AP@0.5r**	**AP@1r**	**AP@0.5r**	**AP@1r**	**AP@0.5r**	**AP@1r**
1BXN	279.81	1.264	▲*	ns	0.802 ± 0.053	0.833 ± 0.054	0.730 ± 0.029	0.809 ± 0.030
7ETM	943.62	1.289	▲**	▲*	0.602 ± 0.027	0.799 ± 0.020	0.509 ± 0.060	0.728 ± 0.075
5OOL	1,797.38	1.388	▲**	▲**	0.806 ± 0.035	0.839 ± 0.035	0.744 ± 0.011	0.786 ± 0.012
3GL1	84.61	1.504	ns	ns	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
1S3X	42.75	1.521	ns	ns	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.001 ± 0.001
7DD9	715.19	1.695	▲**	ns	0.809 ± 0.065	0.827 ± 0.068	0.671 ± 0.089	0.758 ± 0.100
6IGC	2,365.77	1.715	ns	▽**	0.985 ± 0.004	0.991 ± 0.002	0.985 ± 0.002	0.995 ± 0.001
3CF3	270.87	1.849	▲**	▲**	0.583 ± 0.015	0.647 ± 0.015	0.327 ± 0.034	0.455 ± 0.037
6LXV	751.32	1.902	ns	ns	0.949 ± 0.018	0.951 ± 0.017	0.939 ± 0.008	0.950 ± 0.007
5JH9	230.78	2.056	▲**	▲*	0.786 ± 0.056	0.808 ± 0.055	0.674 ± 0.040	0.759 ± 0.042

▲/▽: adding $\mathcal{G}$ significantly improves/degrades performance; ns: no significant difference; *: $p < 0.05$; **: $p < 0.01$.

Table A29: **Effect of $\mathcal{G}$ on unseen proteins in simulated data.**

			$\mathcal{G}$ vs. no $\mathcal{G}$		FullTilt		Without $\mathcal{G}$	
PDB	**Weight (kDa)**	**Axis ratio**	**AP@0.5r**	**AP@1r**	**AP@0.5r**	**AP@1r**	**AP@0.5r**	**AP@1r**
2DF7	987.25	1.116	▽*	ns	0.927 ± 0.042	0.968 ± 0.042	0.968 ± 0.009	0.994 ± 0.009
1E9R	293.47	1.151	▲**	▲*	0.654 ± 0.055	0.711 ± 0.051	0.519 ± 0.031	0.669 ± 0.027
3HHB	64.74	1.269	ns	ns	0.000 ± 0.000	0.003 ± 0.001	0.000 ± 0.000	0.001 ± 0.001
1FPY	629.55	1.271	ns	ns	0.860 ± 0.039	0.864 ± 0.038	0.866 ± 0.056	0.885 ± 0.055
2H12	298.56	1.467	▲**	▲*	0.832 ± 0.015	0.863 ± 0.013	0.671 ± 0.057	0.790 ± 0.058
4D4R	63.66	1.786	ns	ns	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
1JZ8	478.40	2.061	ns	ns	0.850 ± 0.064	0.857 ± 0.065	0.813 ± 0.028	0.835 ± 0.024
6T3E	78.11	2.076	ns	ns	0.000 ± 0.001	0.005 ± 0.003	0.001 ± 0.001	0.002 ± 0.002
1F1B	103.55	2.220	▲**	ns	0.281 ± 0.044	0.486 ± 0.060	0.191 ± 0.020	0.456 ± 0.031
1OAO	316.52	2.936	▲**	ns	0.612 ± 0.044	0.660 ± 0.046	0.525 ± 0.074	0.602 ± 0.075

▲/▽: adding $\mathcal{G}$ significantly improves/degrades performance; ns: no significant difference; *: $p < 0.05$; **: $p < 0.01$.

ratio range, including for the most elongated particle (1OAO, axis ratio 2.94, AP@0.5r: $0.53 \rightarrow 0.61$), whereas performance declines for the most globular particle (2DF7, axis ratio 1.12, AP@0.5r: $0.97 \rightarrow 0.93$).

Spearman correlations between axis ratio and the gain from $\mathcal{G}$, defined as $\Delta\text{AP} = \text{AP}_{\text{FullTilt}} - \text{AP}_{\text{without } \mathcal{G}}$, show no significant association (Table A30). Finally, Spearman correlations between FullTilt's AP and particle properties help characterize its failure cases (Table A31). AP shows no significant association with axis ratio but a strong positive association with molecular weight.

Overall, FullTilt detects non-globular particles, and these results provide no evidence that $\mathcal{G}$ introduces a globular bias among the evaluated protein types. Instead, all evaluated particle types below 100 kDa have near-zero AP, consistent with the small-particle limitation.

Table A30: **Spearman correlations between axis ratio and the AP gain from $\mathcal{G}$.**

Property	**ΔAP@0.5r**	**ΔAP@1r**
Axis ratio	0.155 ($p = 0.515$)	0.203 ($p = 0.390$)

Each entry reports Spearman's ρ and its p-value across the 20 evaluated protein types.

Appendix H.7 Effect of Broader Training-Distribution Coverage

The original FullTilt does not consistently outperform TomoTwin on simulated data. To investigate whether performance can be improved without architectural changes, we broaden the training distribution using an extended $\mathcal{G}$ that incorporates additional noise patterns (white, Poisson, and speckle noise) and geometric primitives (cube and shell). The architecture and optimization hyperparameters remain unchanged. Table A32 reports the results, with statistical comparisons based on Wilcoxon signed-rank tests over 10 trials.

With the same architecture, FullTilt with extended $\mathcal{G}$ significantly outperforms TomoTwin in 6 of the 10 dataset–metric comparisons, shows no significant difference in 3 (with higher point estimates in all three), and performs significantly worse in only 1. TomoTwin retains a higher point estimate

Table A31: **Spearman correlations between FullTilt's AP and particle properties.**

Property	AP@0.5r	AP@1r
Axis ratio	-0.161 ($p = 0.499$)	-0.215 ($p = 0.362$)
Weight	0.844 ($p < 10^{-5}$)	0.858 ($p < 10^{-5}$)

Each entry reports Spearman's ρ and its p-value across the 20 evaluated protein types.

only for mAP@1r on unseen simulated proteins, whereas FullTilt performs significantly better under the stricter mAP@0.5r criterion (0.545 vs. 0.531). These results suggest that training-distribution coverage contributes to the original simulation performance gap and that broadening this distribution can improve performance on simulated data while retaining competitive real-world performance, without modifying the architecture.

Table A32: **FullTilt with extended $\mathcal{G}$.** The architecture and optimization hyperparameters remain unchanged.

	DS-10440		EMPIAR-10304		EMPIAR-10499		Simulation (Seen)		Simulation (Unseen)	
Method	**mAP@0.5r**	**mAP@1r**	**mAP@0.5r**	**mAP@1r**	**mAP@0.5r**	**mAP@1r**	**mAP@0.5r**	**mAP@1r**	**mAP@0.5r**	**mAP@1r**
TomoTwin	0.089 ± 0.021	0.132 ± 0.024	0.189 ± 0.057	0.339 ± 0.063	0.035 ± 0.008	0.113 ± 0.020	0.578 ± 0.018	0.708 ± 0.021	0.531 ± 0.010	0.678 ± 0.021
FullTilt (extended $\mathcal{G}$)	0.163 ± 0.013	0.202 ± 0.014	0.199 ± 0.007	0.400 ± 0.005	0.251 ± 0.004	0.372 ± 0.004	0.694 ± 0.011	0.722 ± 0.010	0.545 ± 0.011	0.577 ± 0.011
FullTilt vs. TomoTwin	▲**	▲**	ns	ns	▲**	▲**	▲**	ns	▲**	▽**

Statistical comparisons use the Wilcoxon signed-rank test over 10 trials. ▲/▽: FullTilt with extended $\mathcal{G}$ performs significantly better/worse than TomoTwin; ns: no significant difference; *: $p < 0.05$; **: $p < 0.01$.

Appendix I Qualitative Detection Results

We provide visual assessments of FullTilt alongside corresponding ground truth annotations and estimations from the state-of-the-art open-set baselines, TomoTwin [9] and ProPicker [11]. To comprehensively demonstrate the capabilities of our framework, FullTilt's 3D detection results are projected back onto the raw 2D tilt-series and additionally visualized across reconstructed tomogram Z-slices. Because TomoTwin and ProPicker operate exclusively on 3D reconstructed volumes, their corresponding detection results are visualized solely on the tomogram Z-slices. All results are derived from the intra-instance, single-prompt configuration. For visual clarity, bounding boxes are uniformly color-coded: FullTilt estimations are displayed in red (with the provided visual prompt in blue), ground truth annotations in lime, TomoTwin estimations in orange, and ProPicker estimations in magenta.

Visualizations for representative instances are presented in the following order: the DS-10440 dataset (Figs. A5–A10), EMPIAR-10304 (Figs. A11–A16), EMPIAR-10499 (Figs. A17–A22), "seen" simulations (Figs. A23–A28), and "unseen" simulations (Figs. A29–A34). As previously noted, while empirical EMPIAR datasets may contain a small fraction of unlabeled particles due to the inherent challenge of obtaining exhaustive ground truth, the annotations remain qualitatively robust for visual comparison.

Appendix I.1 DS-10440 Qualitative Detection Results

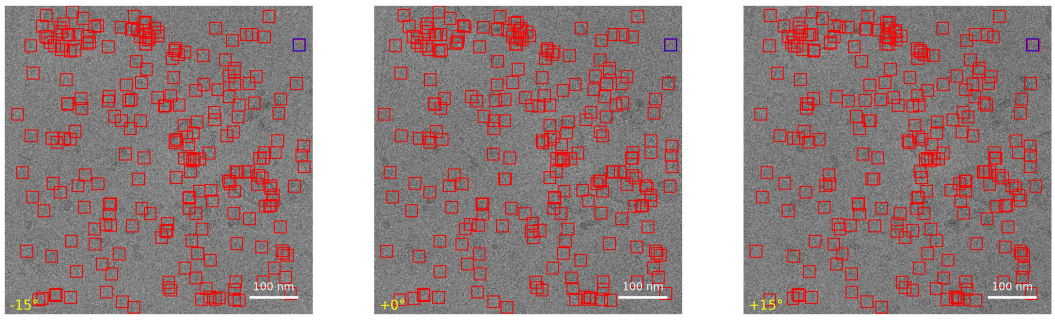


Figure A5: **Projected FullTilt 3D detection results on the 2D tilt-series (DS-10440, `TS_6_6`).** We visualize the zero-shot detection of thyroglobulin particles evaluated under a single-prompt configuration. The reference visual prompt is highlighted by the blue bounding box. To prevent visual clutter, only a representative subset of the projection angles from the full tilt-series is displayed.

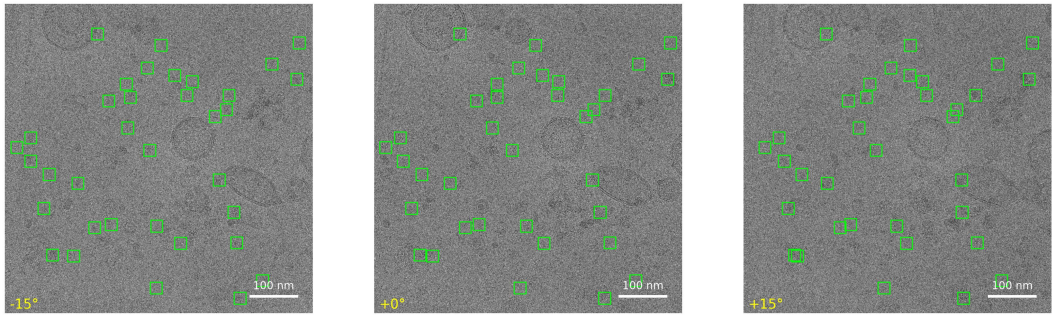


Figure A6: **Projected 3D ground truth annotations on the 2D tilt-series (DS-10440, `TS_6_6`).** Ground truth localizations of thyroglobulin particles are shown. Only a representative subset of the projection angles from the full tilt-series is displayed.

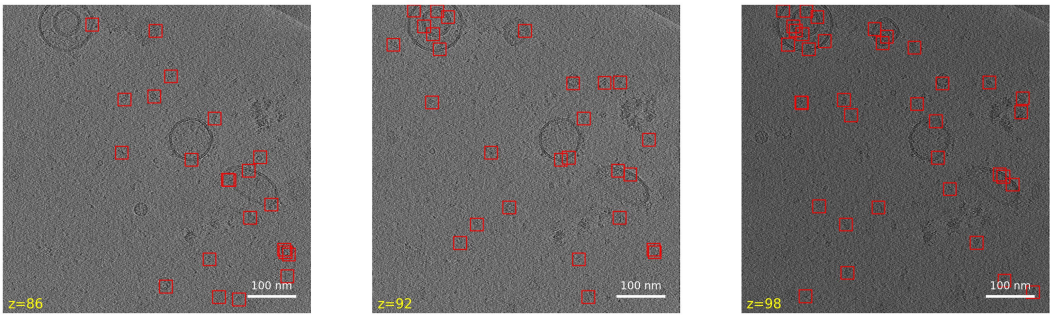


Figure A7: **Tomogram Z-slice visualization of FullTilt detection results (DS-10440, `TS_6_6`).** We visualize the zero-shot detection of thyroglobulin particles evaluated under a single-prompt configuration. For each displayed xy-plane at depth z, we render only the estimated bounding boxes whose center coordinates fall within a local depth window defined by the particle's radius r (i.e., $z \pm r$). A representative subset of the tomogram Z-slices is shown for visual clarity.

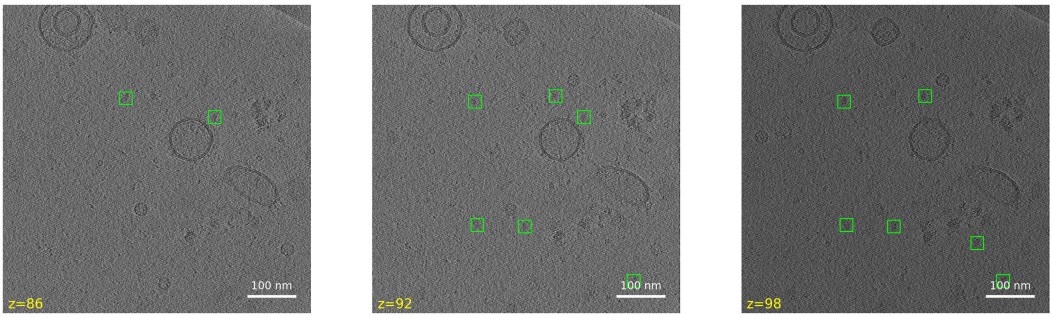


Figure A8: **Tomogram Z-slice visualization of 3D ground truth (DS-10440, `TS_6_6`).** Ground truth localizations of thyroglobulin particles are shown. For each displayed xy-plane at depth z, we render only the bounding boxes whose center coordinates fall within a local depth window defined by the particle's radius r (i.e., $z \pm r$). A representative subset of the tomogram Z-slices is shown for visual clarity.

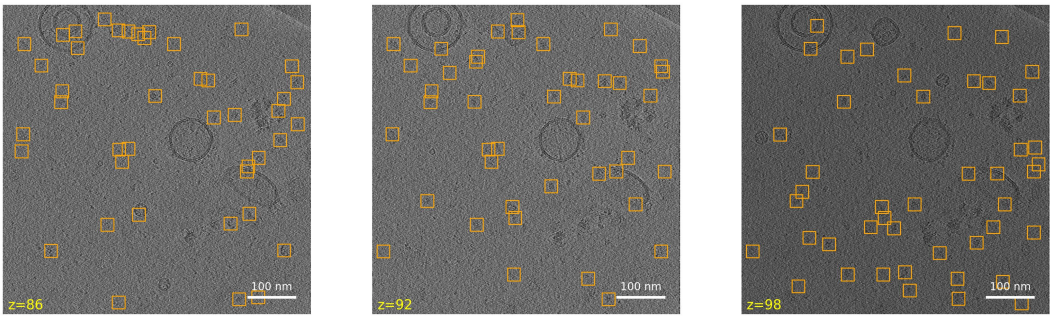


Figure A9: **Tomogram Z-slice visualization of TomoTwin detection results (DS-10440, `TS_6_6`).** Estimated localizations of thyroglobulin particles are shown. For each displayed xy-plane at depth z, we render only the bounding boxes whose center coordinates fall within a local depth window defined by the particle's radius r (i.e., $z \pm r$). A representative subset of the tomogram Z-slices is shown for visual clarity.

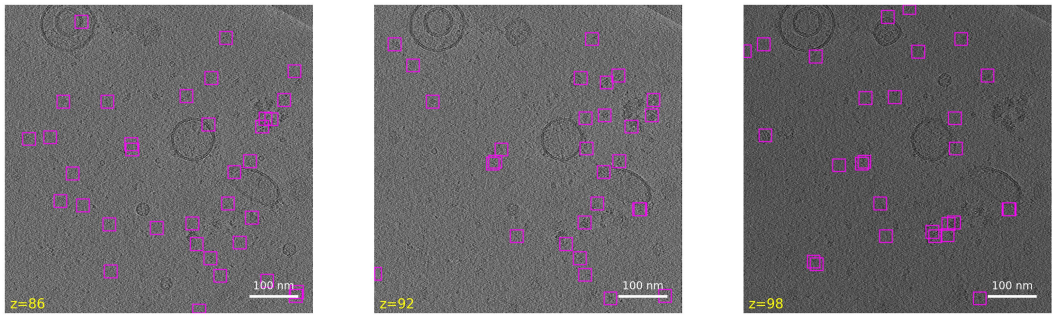


Figure A10: **Tomogram Z-slice visualization of ProPicker detection results (DS-10440,** `TS_6_6`**).** Estimated localizations of thyroglobulin particles are shown. For each displayed xy-plane at depth z, we render only the bounding boxes whose center coordinates fall within a local depth window defined by the particle's radius r (i.e., $z \pm r$). A representative subset of the tomogram Z-slices is shown for visual clarity.

Appendix I.2 EMPIAR-10304 Qualitative Detection Results

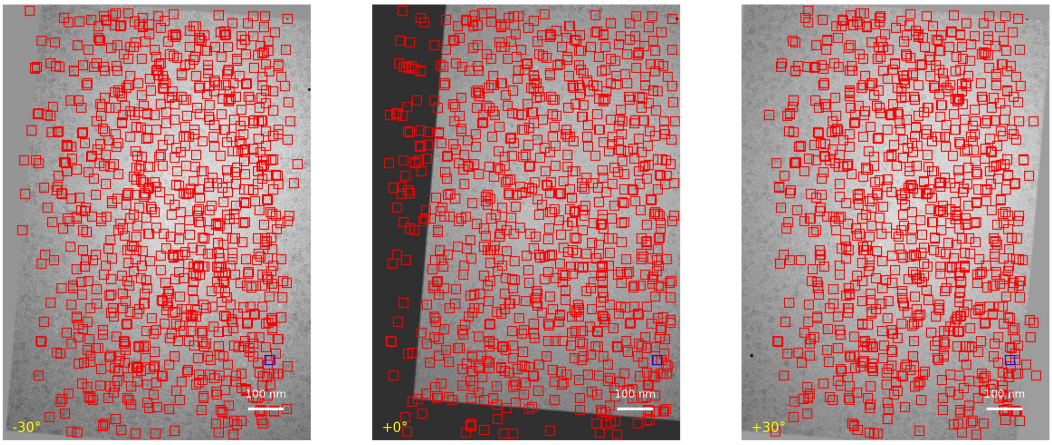


Figure A11: **Projected FullTilt 3D detection results on the 2D tilt-series (EMPIAR-10304, `tilt1`).** We visualize the zero-shot detection of *Escherichia coli* 70S ribosomes evaluated under a single-prompt configuration. The reference visual prompt is highlighted by the blue bounding box. To prevent visual clutter, only a representative subset of the projection angles from the full tilt-series is displayed.

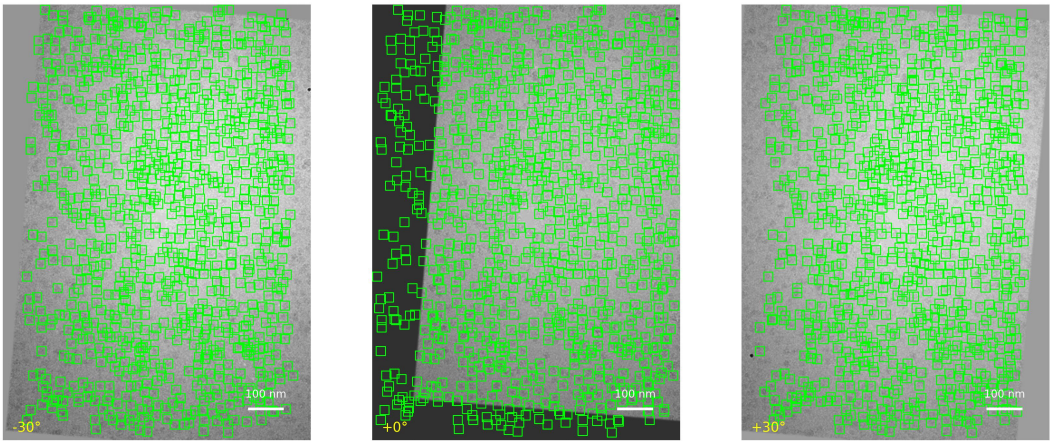


Figure A12: **Projected 3D ground truth annotations on the 2D tilt-series (EMPIAR-10304, `tilt1`).** Ground truth localizations of *Escherichia coli* 70S ribosomes are shown. Only a representative subset of the projection angles from the full tilt-series is displayed. Note that while a small fraction of particles may remain unlabeled due to the prohibitive difficulty of exhaustive 3D annotation, these empirical ground truth labels remain qualitatively robust.

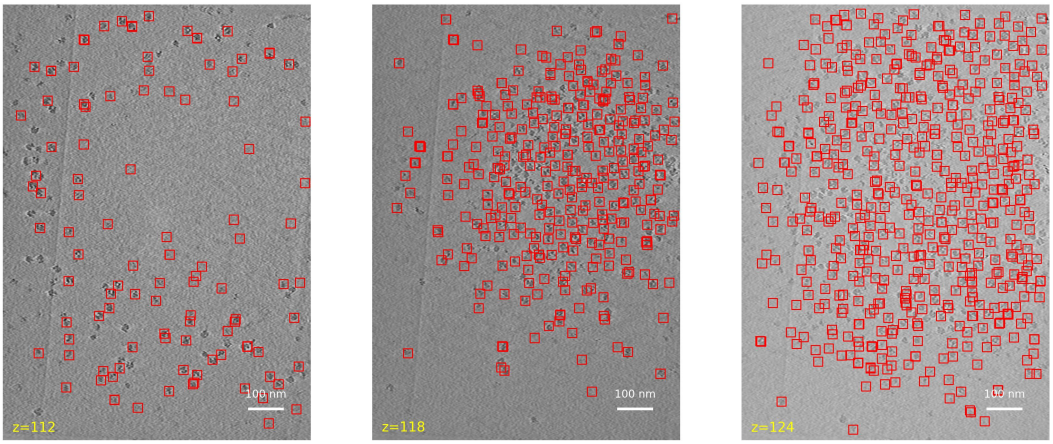


Figure A13: **Tomogram Z-slice visualization of FullTilt detection results (EMPIAR-10304, `tilt1`).** We visualize the zero-shot detection of *Escherichia coli* 70S ribosomes evaluated under a single-prompt configuration. For each displayed xy-plane at depth z, we render only the estimated bounding boxes whose center coordinates fall within a local depth window defined by the particle's radius r (i.e., $z \pm r$). A representative subset of the tomogram Z-slices is shown for visual clarity.

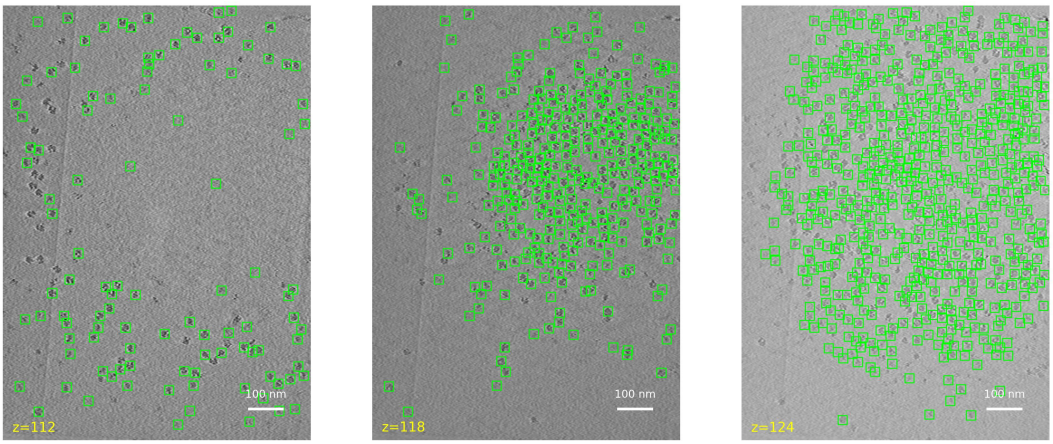


Figure A14: **Tomogram Z-slice visualization of 3D ground truth (EMPIAR-10304, `tilt1`).** Ground truth localizations of *Escherichia coli* 70S ribosomes are shown. For each displayed xy-plane at depth z, we render only the bounding boxes whose center coordinates fall within a local depth window defined by the particle's radius r (i.e., $z \pm r$). A representative subset of the tomogram Z-slices is shown for visual clarity. Note that while a small fraction of particles may remain unlabeled due to the prohibitive difficulty of exhaustive 3D annotation, these empirical ground truth labels remain qualitatively robust.

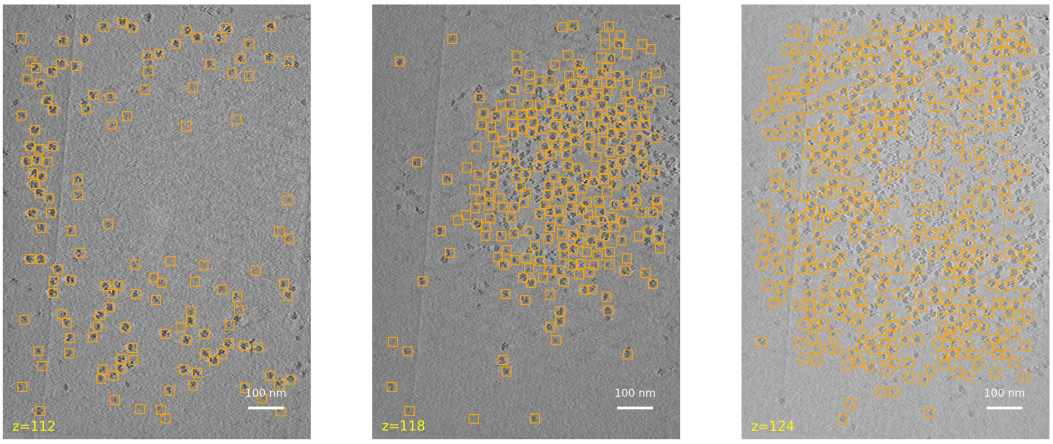


Figure A15: **Tomogram Z-slice visualization of TomoTwin detection results (EMPIAR-10304, `tilt1`).** Estimated localizations of *Escherichia coli* 70S ribosomes are shown. For each displayed xy-plane at depth z, we render only the bounding boxes whose center coordinates fall within a local depth window defined by the particle's radius r (i.e., $z \pm r$). A representative subset of the tomogram Z-slices is shown for visual clarity.

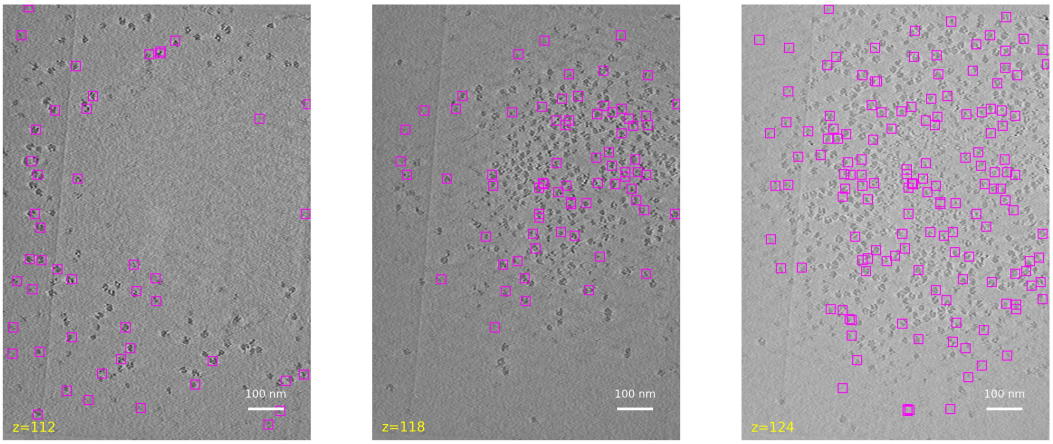


Figure A16: **Tomogram Z-slice visualization of ProPicker detection results (EMPIAR-10304, `tilt1`).** Estimated localizations of *Escherichia coli* 70S ribosomes are shown. For each displayed xy-plane at depth z, we render only the bounding boxes whose center coordinates fall within a local depth window defined by the particle's radius r (i.e., $z \pm r$). A representative subset of the tomogram Z-slices is shown for visual clarity.

Appendix I.3 EMPIAR-10499 Qualitative Detection Results

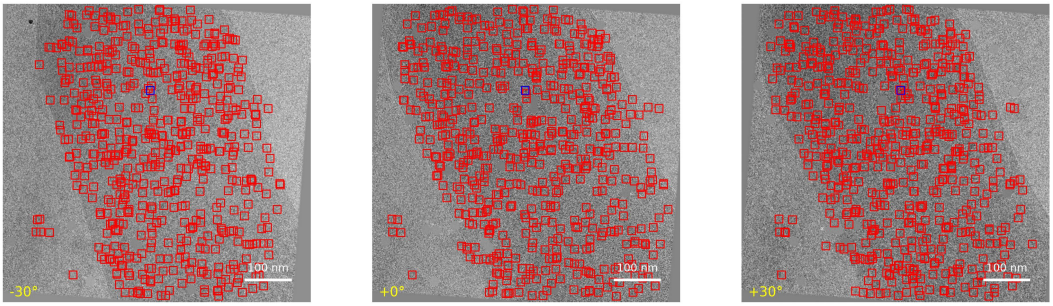


Figure A17: **Projected FullTilt 3D detection results on the 2D tilt-series (EMPIAR-10499, `TS_01`).** We visualize the zero-shot detection of *Mycoplasma pneumoniae* ribosomes evaluated under a single-prompt configuration. The reference visual prompt is highlighted by the blue bounding box. To prevent visual clutter, only a representative subset of the projection angles from the full tilt-series is displayed.

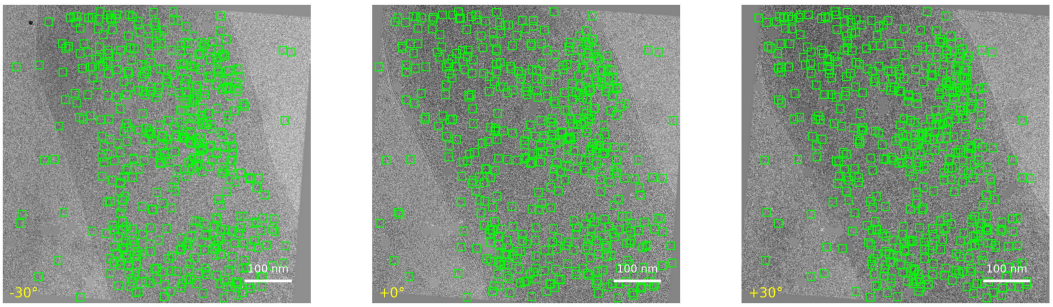


Figure A18: **Projected 3D ground truth annotations on the 2D tilt-series (EMPIAR-10499, `TS_01`).** Ground truth localizations of *Mycoplasma pneumoniae* ribosomes are shown. Only a representative subset of the projection angles from the full tilt-series is displayed. Note that while a small fraction of particles may remain unlabeled due to the prohibitive difficulty of exhaustive 3D annotation, these empirical ground truth labels remain qualitatively robust.

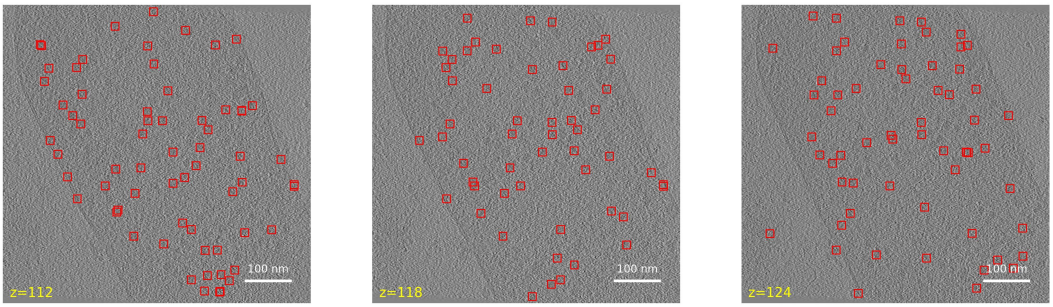


Figure A19: **Tomogram Z-slice visualization of FullTilt detection results (EMPIAR-10499, `TS_01`).** We visualize the zero-shot detection of *Mycoplasma pneumoniae* ribosomes evaluated under a single-prompt configuration. For each displayed xy-plane at depth z, we render only the estimated bounding boxes whose center coordinates fall within a local depth window defined by the particle's radius r (i.e., $z \pm r$). A representative subset of the tomogram Z-slices is shown for visual clarity.

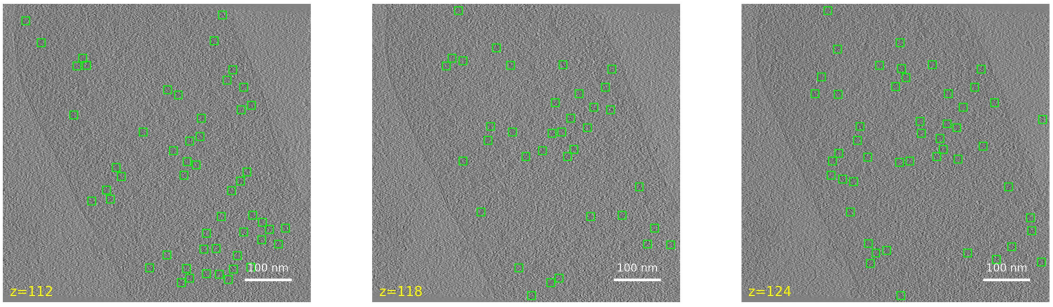


Figure A20: **Tomogram Z-slice visualization of 3D ground truth (EMPIAR-10499, `TS_01`).** Ground truth localizations of *Mycoplasma pneumoniae* ribosomes are shown. For each displayed xy-plane at depth z, we render only the bounding boxes whose center coordinates fall within a local depth window defined by the particle's radius r (i.e., $z \pm r$). A representative subset of the tomogram Z-slices is shown for visual clarity. Note that while a small fraction of particles may remain unlabeled due to the prohibitive difficulty of exhaustive 3D annotation, these empirical ground truth labels remain qualitatively robust.

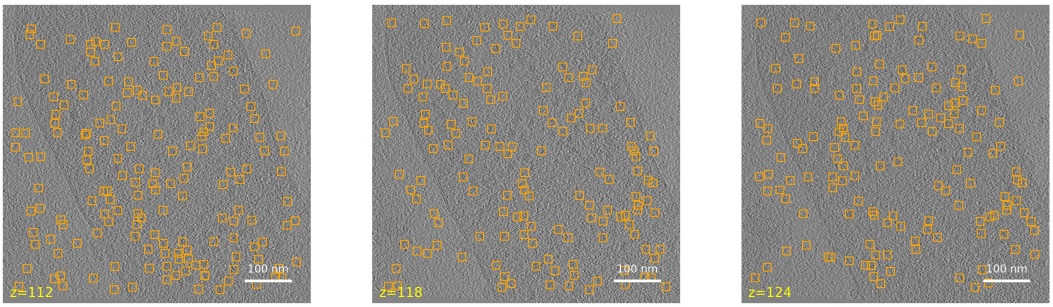


Figure A21: **Tomogram Z-slice visualization of TomoTwin detection results (EMPIAR-10499, `TS_01`).** Estimated localizations of *Mycoplasma pneumoniae* ribosomes are shown. For each displayed xy-plane at depth z, we render only the bounding boxes whose center coordinates fall within a local depth window defined by the particle's radius r (i.e., $z \pm r$). A representative subset of the tomogram Z-slices is shown for visual clarity.

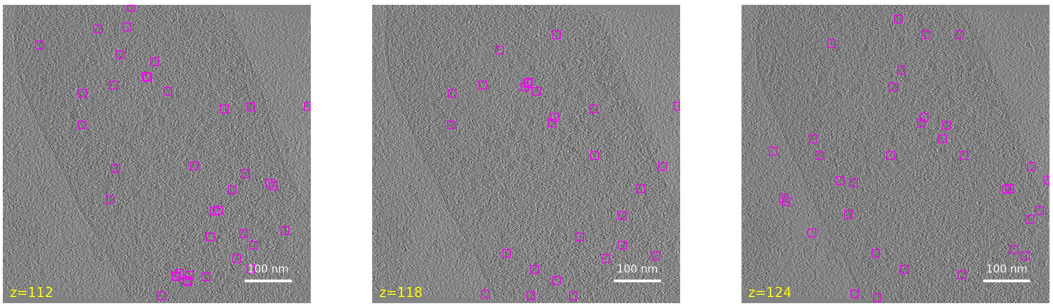


Figure A22: **Tomogram Z-slice visualization of ProPicker detection results (EMPIAR-10499,** `TS_01`**).** Estimated localizations of *Mycoplasma pneumoniae* ribosomes are shown. For each displayed xy-plane at depth z, we render only the bounding boxes whose center coordinates fall within a local depth window defined by the particle's radius r (i.e., $z \pm r$). A representative subset of the tomogram Z-slices is shown for visual clarity.

Appendix I.4 "Seen" Simulation Qualitative Detection Results

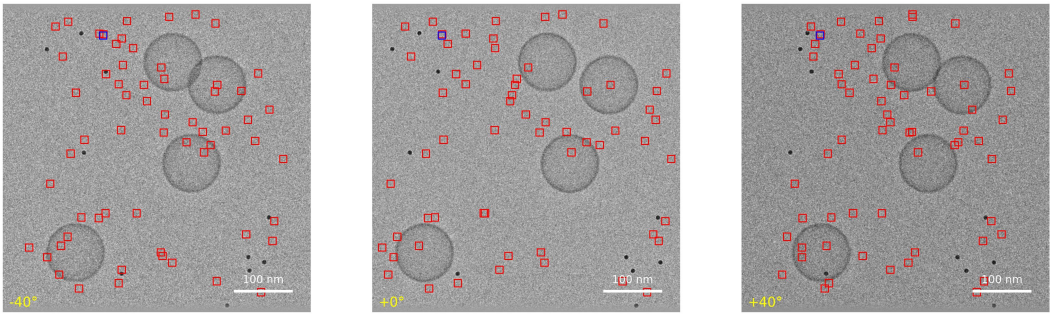


Figure A23: **Projected FullTilt 3D detection results on the 2D tilt-series (seen simulation, `TS_4_0`).** We visualize the detection of `5JH9` particles evaluated under a single-prompt configuration. The reference visual prompt is highlighted by the blue bounding box. To prevent visual clutter, only a representative subset of the projection angles from the full tilt-series is displayed.

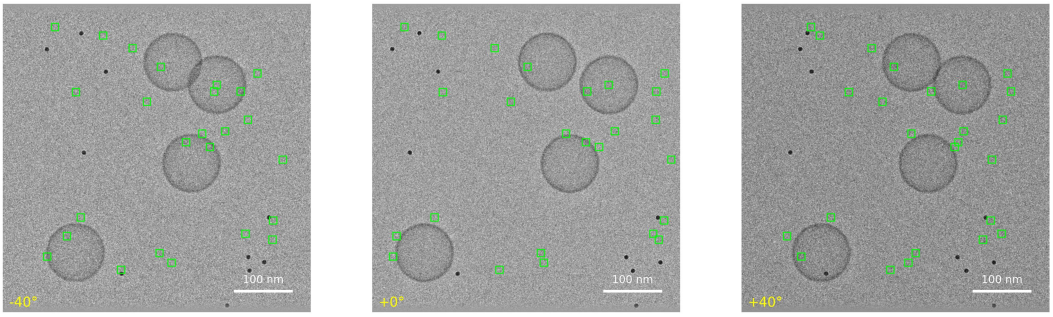


Figure A24: **Projected 3D ground truth annotations on the 2D tilt-series (seen simulation, `TS_4_0`).** Ground truth localizations of `5JH9` particles are shown. Only a representative subset of the projection angles from the full tilt-series is displayed.

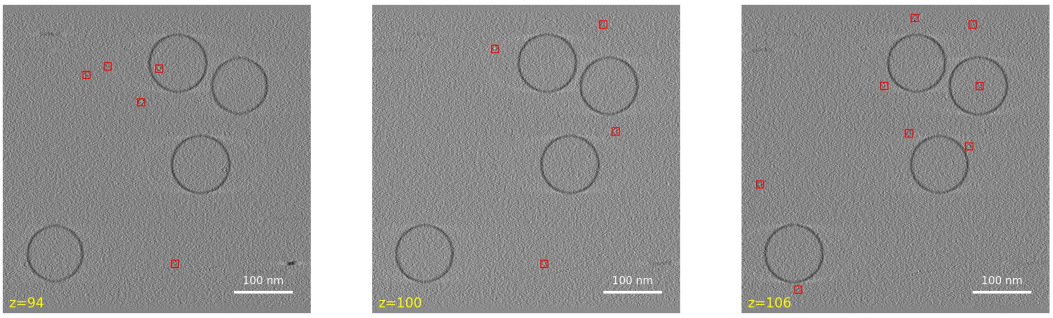


Figure A25: **Tomogram Z-slice visualization of FullTilt detection results ("seen" simulation, `TS_4_0`).** We visualize the detection of `5JH9` particles evaluated under a single-prompt configuration. For each displayed xy-plane at depth z, we render only the estimated bounding boxes whose center coordinates fall within a local depth window defined by the particle's radius r (i.e., $z \pm r$). A representative subset of the tomogram Z-slices is shown for visual clarity.

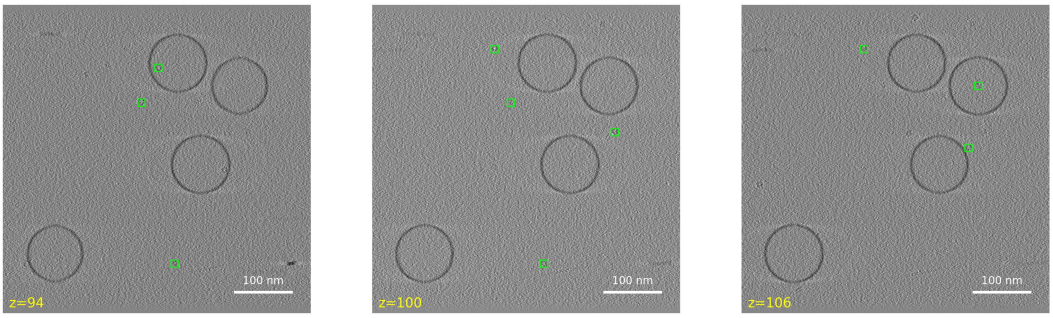


Figure A26: **Tomogram Z-slice visualization of 3D ground truth ("seen" simulation, `TS_4_0`).** Ground truth localizations of `5JH9` particles are shown. For each displayed xy-plane at depth z, we render only the bounding boxes whose center coordinates fall within a local depth window defined by the particle's radius r (i.e., $z \pm r$). A representative subset of the tomogram Z-slices is shown for visual clarity.

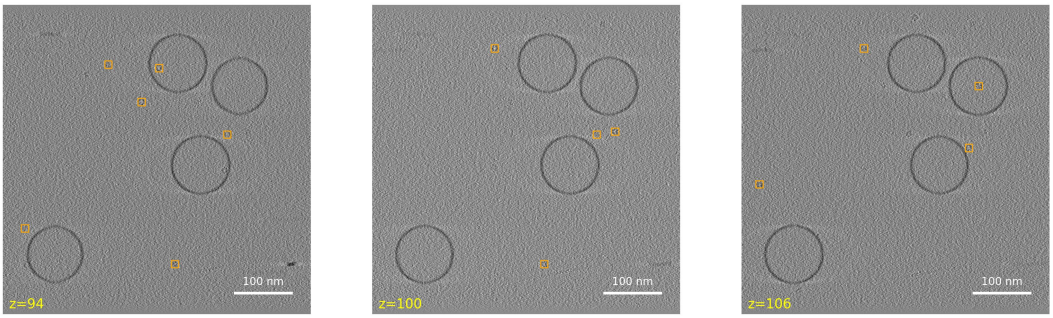


Figure A27: **Tomogram Z-slice visualization of TomoTwin detection results ("seen" simulation, `TS_4_0`).** Estimated localizations of `5JH9` particles are shown. For each displayed xy-plane at depth z, we render only the bounding boxes whose center coordinates fall within a local depth window defined by the particle's radius r (i.e., $z \pm r$). A representative subset of the tomogram Z-slices is shown for visual clarity.

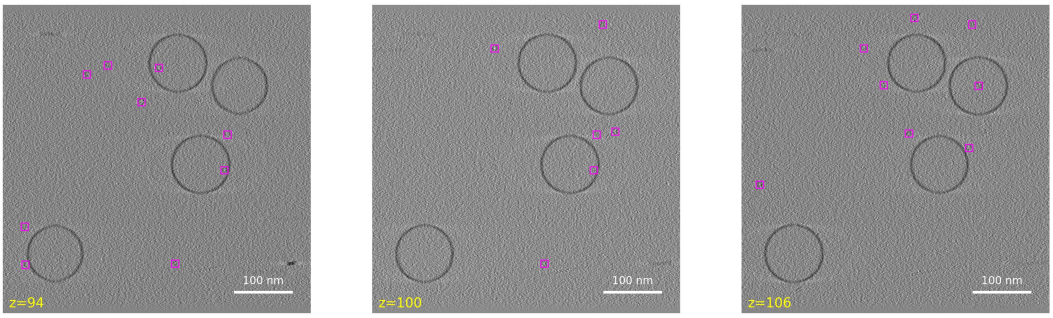


Figure A28: **Tomogram Z-slice visualization of ProPicker detection results ("seen" simulation, `TS_4_0`).** Estimated localizations of `5JH9` particles are shown. For each displayed xy-plane at depth z, we render only the bounding boxes whose center coordinates fall within a local depth window defined by the particle's radius r (i.e., $z \pm r$). A representative subset of the tomogram Z-slices is shown for visual clarity.

Appendix I.5 "Unseen" Simulation Qualitative Detection Results

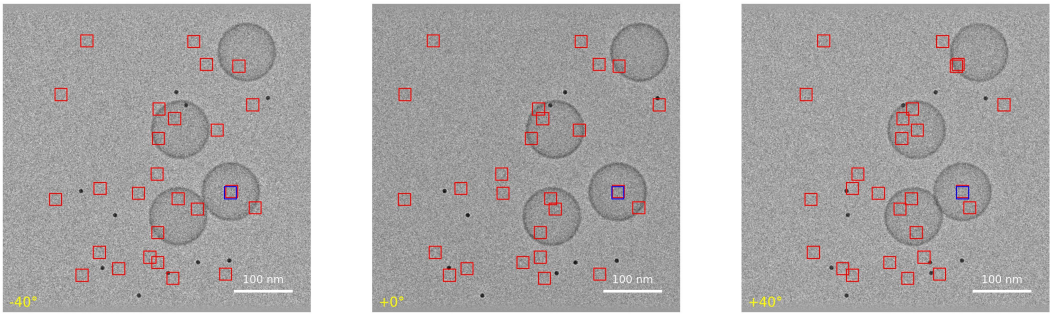


Figure A29: **Projected FullTilt 3D detection results on the 2D tilt-series ("unseen" simulation, `TS_4_1`).** We visualize the zero-shot detection of `2DF7` particles evaluated under a single-prompt configuration. The reference visual prompt is highlighted by the blue bounding box. To prevent visual clutter, only a representative subset of the projection angles from the full tilt-series is displayed.

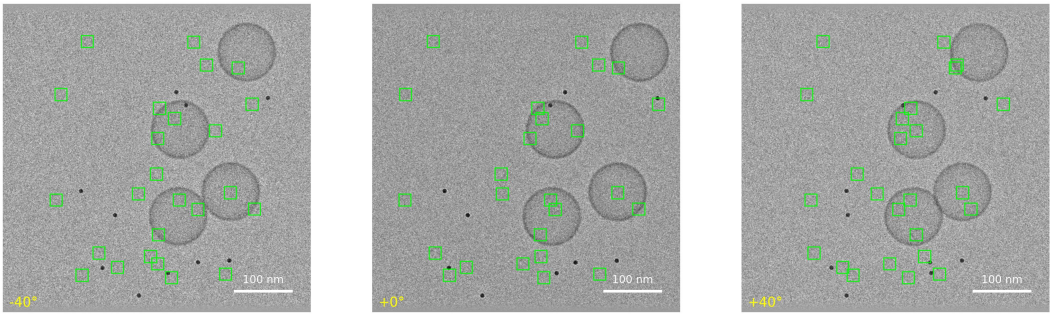


Figure A30: **Projected 3D ground truth annotations on the 2D tilt-series ("unseen" simulation, `TS_4_1`).** Ground truth localizations of `2DF7` particles are shown. Only a representative subset of the projection angles from the full tilt-series is displayed.

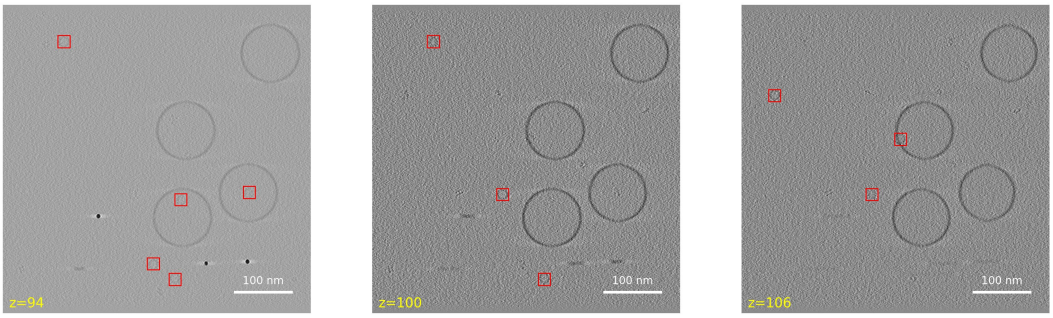


Figure A31: **Tomogram Z-slice visualization of FullTilt detection results ("unseen" simulation, `TS_4_1`).** We visualize the zero-shot detection of `2DF7` particles evaluated under a single-prompt configuration. For each displayed xy-plane at depth z, we render only the estimated bounding boxes whose center coordinates fall within a local depth window defined by the particle's radius r (i.e., $z \pm r$). A representative subset of the tomogram Z-slices is shown for visual clarity.

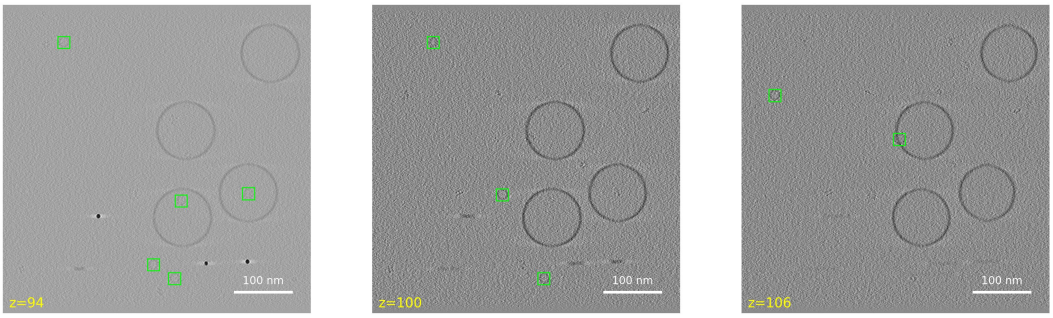


Figure A32: **Tomogram Z-slice visualization of 3D ground truth ("unseen" simulation, `TS_4_1`).** Ground truth localizations of `2DF7` particles are shown. For each displayed xy-plane at depth z, we render only the bounding boxes whose center coordinates fall within a local depth window defined by the particle's radius r (i.e., $z \pm r$). A representative subset of the tomogram Z-slices is shown for visual clarity.

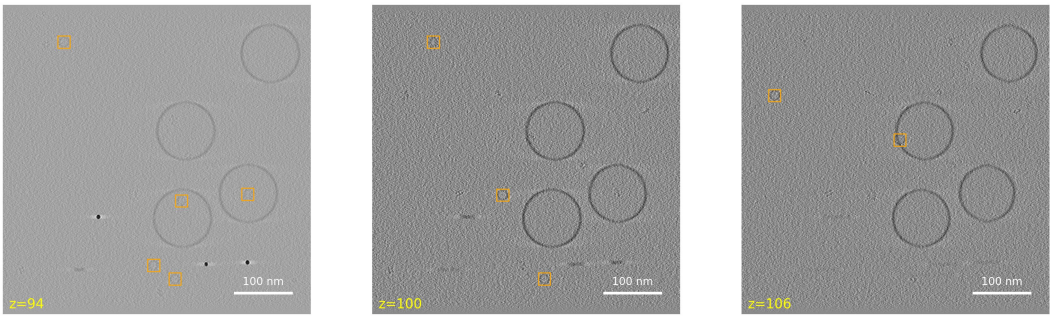


Figure A33: **Tomogram Z-slice visualization of TomoTwin detection results ("seen" simulation, `TS_4_1`).** Estimated localizations of `2DF7` particles are shown. For each displayed xy-plane at depth z, we render only the bounding boxes whose center coordinates fall within a local depth window defined by the particle's radius r (i.e., $z \pm r$). A representative subset of the tomogram Z-slices is shown for visual clarity.

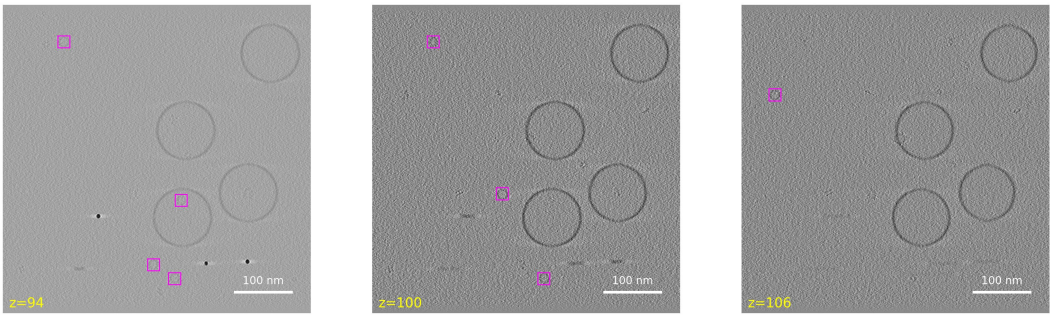


Figure A34: **Tomogram Z-slice visualization of ProPicker detection results ("seen" simulation, `TS_4_1`).** Estimated localizations of `2DF7` particles are shown. For each displayed xy-plane at depth z, we render only the bounding boxes whose center coordinates fall within a local depth window defined by the particle's radius r (i.e., $z \pm r$). A representative subset of the tomogram Z-slices is shown for visual clarity.

Appendix J Comprehensive Ablation Study

Appendix J.1 Additional Results on the Number of Tilt Images

We evaluate the impact of utilizing fewer tilt images by systematically testing retention rates of 100%, 50%, 25%, and 12.5%. These sparse subsets are generated by progressively doubling the angular stride between sampled projections. The additional results are detailed in Tables A33 (1 prompt), A34 (2 prompts), and A35 (4 prompts) for the real-world datasets, and Tables A36 (1 prompt), A37 (2 prompts), and A38 (4 prompts) for the simulation datasets. As anticipated, reducing the density of the tilt-series leads to a consistent degradation in detection performance, which is accompanied by proportional gains in inference speed and reduced VRAM consumption.

Table A33: **Ablation study on the number of tilt images (real-world datasets; 1 prompt).** Reducing the input sequence inherently decreases runtime and memory at the cost of overall detection performance.

	DS-10440				EMPIAR-10304				EMPIAR-10499			
# Tilt Images	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**
100%	0.118 ± 0.013	0.148 ± 0.013	0.499	3430	0.199 ± 0.004	0.437 ± 0.002	0.229	2410	0.159 ± 0.004	0.288 ± 0.004	0.389	3410
50%	0.100 ± 0.013	0.133 ± 0.011	0.246	1860	0.184 ± 0.002	0.427 ± 0.002	0.118	1380	0.127 ± 0.006	0.260 ± 0.007	0.202	1950
25%	0.085 ± 0.010	0.120 ± 0.011	0.153	1280	0.157 ± 0.003	0.391 ± 0.003	0.079	860	0.060 ± 0.005	0.165 ± 0.008	0.116	1220
12.5%	0.038 ± 0.007	0.073 ± 0.014	0.087	700	0.084 ± 0.003	0.268 ± 0.003	0.069	600	0.010 ± 0.001	0.048 ± 0.003	0.083	790

T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

Table A34: **Ablation study on the number of tilt images (real-world datasets; 2 prompts).** Reducing the input sequence inherently decreases runtime and memory at the cost of overall detection performance.

	DS-10440				EMPIAR-10304				EMPIAR-10499			
# Tilt Images	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**
100%	0.129 ± 0.010	0.161 ± 0.011	0.503	3430	0.200 ± 0.004	0.437 ± 0.003	0.230	2410	0.169 ± 0.006	0.302 ± 0.009	0.394	3410
50%	0.114 ± 0.007	0.150 ± 0.008	0.246	1860	0.183 ± 0.003	0.426 ± 0.002	0.119	1380	0.134 ± 0.006	0.272 ± 0.008	0.205	1950
25%	0.097 ± 0.006	0.134 ± 0.006	0.153	1280	0.156 ± 0.002	0.390 ± 0.002	0.082	860	0.065 ± 0.003	0.177 ± 0.007	0.116	1220
12.5%	0.044 ± 0.006	0.082 ± 0.009	0.088	700	0.085 ± 0.002	0.270 ± 0.004	0.072	600	0.011 ± 0.001	0.050 ± 0.002	0.083	792

T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

Table A35: **Ablation study on the number of tilt images (real-world datasets; 4 prompts).** Reducing the input sequence inherently decreases runtime and memory at the cost of overall detection performance.

	DS-10440				EMPIAR-10304				EMPIAR-10499			
# Tilt Images	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**
100%	0.133 ± 0.007	0.180 ± 0.007	0.510	3430	0.201 ± 0.004	0.439 ± 0.001	0.230	2410	0.173 ± 0.002	0.309 ± 0.003	0.394	3410
50%	0.117 ± 0.007	0.157 ± 0.007	0.244	1860	0.186 ± 0.003	0.429 ± 0.002	0.122	1380	0.138 ± 0.002	0.281 ± 0.004	0.203	1950
25%	0.099 ± 0.006	0.139 ± 0.005	0.150	1280	0.155 ± 0.002	0.389 ± 0.003	0.083	860	0.069 ± 0.004	0.185 ± 0.008	0.115	1220
12.5%	0.046 ± 0.006	0.086 ± 0.010	0.081	700	0.083 ± 0.002	0.269 ± 0.001	0.077	600	0.011 ± 0.001	0.052 ± 0.003	0.082	792

T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

Table A36: **Ablation study on the number of tilt images (simulation datasets; 1 prompt).** Reducing the input sequence inherently decreases runtime and memory at the cost of overall detection performance.

	Simulation (Seen)				Simulation (Unseen)			
# Tilt Images	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**
100%	0.632 ± 0.006	0.670 ± 0.007	0.682	5160	0.502 ± 0.012	0.542 ± 0.013	0.694	5160
50%	0.567 ± 0.018	0.639 ± 0.019	0.329	2820	0.426 ± 0.023	0.487 ± 0.021	0.334	2820
25%	0.377 ± 0.018	0.532 ± 0.022	0.163	1570	0.272 ± 0.025	0.380 ± 0.031	0.167	1570
12.5%	0.151 ± 0.029	0.244 ± 0.037	0.093	949	0.069 ± 0.018	0.135 ± 0.025	0.092	949

T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

Table A37: **Ablation study on the number of tilt images (simulation datasets; 2 prompts).** Reducing the input sequence inherently decreases runtime and memory at the cost of overall detection performance.

	Simulation (Seen)				Simulation (Unseen)			
# Tilt Images	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**
100%	0.631 ± 0.008	0.670 ± 0.007	0.685	5160	0.513 ± 0.010	0.552 ± 0.011	0.698	5160
50%	0.577 ± 0.011	0.648 ± 0.014	0.329	2820	0.452 ± 0.013	0.511 ± 0.012	0.334	2820
25%	0.400 ± 0.010	0.559 ± 0.014	0.163	1570	0.295 ± 0.015	0.408 ± 0.017	0.162	1570
12.5%	0.192 ± 0.021	0.311 ± 0.033	0.090	949	0.101 ± 0.018	0.192 ± 0.025	0.094	949

T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

Table A38: **Ablation study on the number of tilt images (simulation datasets; 4 prompts).** Reducing the input sequence inherently decreases runtime and memory at the cost of overall detection performance.

	Simulation (Seen)				Simulation (Unseen)			
# Tilt Images	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**
100%	0.627 ± 0.008	0.662 ± 0.007	0.687	5160	0.513 ± 0.004	0.553 ± 0.005	0.699	5160
50%	0.574 ± 0.007	0.644 ± 0.008	0.331	2820	0.463 ± 0.003	0.521 ± 0.004	0.334	2820
25%	0.408 ± 0.007	0.562 ± 0.009	0.166	1570	0.324 ± 0.009	0.436 ± 0.007	0.165	1570
12.5%	0.222 ± 0.010	0.354 ± 0.018	0.094	949	0.129 ± 0.016	0.239 ± 0.025	0.089	949

T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

Appendix J.2 Number of Projections per Prompt

To demonstrate the flexibility of our visual prompt encoder ($\mathcal{E}_{\text{Prompt}}$), we evaluate scenarios where visual prompts are not provided across the entire tilt-series. Specifically, we test retention settings where prompts are present in 100%, 50%, 25%, and 12.5% of the tilt images, alongside an extreme "single-image prompt" setting. The comprehensive results across the 1-, 2-, and 4-prompt configurations are detailed in Tables A39, A40, and A41 for the real-world datasets, and Tables A42, A43, and A44 for the simulation datasets. Remarkably, FullTilt remains highly robust even in the most extreme scenario—providing just a single visual prompt on a single tilt image (specifically, the 0° projection)—incurring only a marginal degradation in detection performance.

Table A39: **Ablation study on the number of projections per prompt (real-world datasets; 1 prompt).** Evaluation of FullTilt's robustness when prompts are provided on only a subset of the tilt-series images.

	DS-10440				EMPIAR-10304				EMPIAR-10499			
# Projections per Prompt	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**
100%	0.118 ± 0.013	0.148 ± 0.013	0.499	3430	0.199 ± 0.004	0.437 ± 0.002	0.229	2410	0.159 ± 0.004	0.288 ± 0.004	0.389	3410
50%	0.111 ± 0.008	0.142 ± 0.009	0.514	3430	0.199 ± 0.003	0.437 ± 0.002	0.223	2420	0.162 ± 0.005	0.290 ± 0.005	0.392	3410
25%	0.107 ± 0.013	0.136 ± 0.014	0.513	3430	0.199 ± 0.004	0.437 ± 0.003	0.223	2420	0.158 ± 0.006	0.285 ± 0.007	0.391	3410
12.5%	0.108 ± 0.009	0.137 ± 0.010	0.512	3430	0.197 ± 0.005	0.436 ± 0.003	0.224	2420	0.158 ± 0.005	0.285 ± 0.006	0.391	3410
Only One	0.106 ± 0.014	0.134 ± 0.016	0.513	3430	0.190 ± 0.004	0.430 ± 0.004	0.224	2420	0.155 ± 0.009	0.278 ± 0.011	0.390	3410

T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

Table A40: **Ablation study on the number of projections per prompt (real-world datasets; 2 prompts).** Evaluation of FullTilt's robustness when prompts are provided on only a subset of the tilt-series images.

	DS-10440				EMPIAR-10304				EMPIAR-10499			
# Projections per Prompt	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**
100%	0.129 ± 0.010	0.161 ± 0.011	0.503	3430	0.200 ± 0.004	0.437 ± 0.003	0.230	2410	0.169 ± 0.006	0.302 ± 0.009	0.394	3410
50%	0.124 ± 0.009	0.157 ± 0.009	0.513	3430	0.199 ± 0.003	0.437 ± 0.003	0.226	2420	0.168 ± 0.003	0.301 ± 0.004	0.392	3410
25%	0.123 ± 0.011	0.155 ± 0.012	0.514	3430	0.199 ± 0.003	0.438 ± 0.003	0.225	2420	0.166 ± 0.006	0.297 ± 0.009	0.391	3410
12.5%	0.126 ± 0.012	0.156 ± 0.013	0.514	3430	0.198 ± 0.003	0.436 ± 0.002	0.225	2420	0.167 ± 0.005	0.300 ± 0.007	0.390	3410
Only One	0.108 ± 0.011	0.137 ± 0.009	0.513	3430	0.190 ± 0.005	0.429 ± 0.006	0.225	2420	0.164 ± 0.006	0.293 ± 0.008	0.390	3410

T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

Table A41: **Ablation study on the number of projections per prompt (real-world datasets; 4 prompts).** Evaluation of FullTilt's robustness when prompts are provided on only a subset of the tilt-series images.

	DS-10440				EMPIAR-10304				EMPIAR-10499			
# Projections per Prompt	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**
100%	0.133 ± 0.007	0.180 ± 0.007	0.510	3430	0.201 ± 0.004	0.439 ± 0.001	0.230	2410	0.173 ± 0.002	0.309 ± 0.003	0.394	3410
50%	0.130 ± 0.007	0.165 ± 0.006	0.514	3430	0.199 ± 0.002	0.438 ± 0.002	0.227	2420	0.175 ± 0.003	0.311 ± 0.004	0.392	3410
25%	0.133 ± 0.005	0.167 ± 0.005	0.514	3430	0.199 ± 0.002	0.438 ± 0.002	0.226	2420	0.172 ± 0.004	0.307 ± 0.005	0.392	3410
12.5%	0.130 ± 0.006	0.160 ± 0.007	0.514	3430	0.197 ± 0.004	0.437 ± 0.002	0.226	2420	0.170 ± 0.005	0.306 ± 0.007	0.391	3410
Only One	0.117 ± 0.009	0.146 ± 0.009	0.514	3430	0.191 ± 0.004	0.433 ± 0.004	0.225	2420	0.175 ± 0.006	0.306 ± 0.006	0.392	3410

T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

Table A42: **Ablation study on reducing the number of prompted tilt images (simulation datasets; 1 prompt).** Evaluation of FullTilt's robustness when prompts are provided on only a subset of the tilt-series images.

	Simulation (Seen)				Simulation (Unseen)			
# Projections per Prompt	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**
100%	0.632 ± 0.006	0.670 ± 0.007	0.682	5160	0.502 ± 0.012	0.542 ± 0.013	0.694	5160
50%	0.625 ± 0.012	0.665 ± 0.012	0.682	5160	0.500 ± 0.008	0.543 ± 0.008	0.694	5160
25%	0.629 ± 0.013	0.666 ± 0.014	0.681	5160	0.483 ± 0.016	0.523 ± 0.018	0.694	5160
12.5%	0.609 ± 0.019	0.645 ± 0.019	0.680	5160	0.486 ± 0.025	0.525 ± 0.025	0.693	5160
Only One	0.572 ± 0.025	0.608 ± 0.026	0.661	5160	0.431 ± 0.017	0.467 ± 0.015	0.693	5160

T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

Table A43: **Ablation study on reducing the number of prompted tilt images (simulation datasets; 2 prompts).** Evaluation of FullTilt's robustness when prompts are provided on only a subset of the tilt-series images.

	Simulation (Seen)				Simulation (Unseen)			
# Projections per Prompt	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**
100%	0.631 ± 0.008	0.670 ± 0.007	0.685	5160	0.513 ± 0.010	0.552 ± 0.011	0.698	5160
50%	0.630 ± 0.010	0.668 ± 0.010	0.682	5160	0.510 ± 0.011	0.551 ± 0.011	0.695	5160
25%	0.626 ± 0.006	0.662 ± 0.006	0.681	5160	0.510 ± 0.009	0.549 ± 0.008	0.695	5160
12.5%	0.628 ± 0.009	0.663 ± 0.008	0.681	5160	0.505 ± 0.013	0.543 ± 0.013	0.694	5160
Only One	0.594 ± 0.013	0.629 ± 0.015	0.681	5160	0.475 ± 0.027	0.513 ± 0.027	0.692	5160

T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

Table A44: **Ablation study on reducing the number of prompted tilt images (simulation datasets; 4 prompts).** Evaluation of FullTilt's robustness when prompts are provided on only a subset of the tilt-series images.

	Simulation (Seen)				Simulation (Unseen)			
# Projections per Prompt	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**
100%	0.627 ± 0.008	0.662 ± 0.007	0.687	5160	0.513 ± 0.004	0.553 ± 0.005	0.699	5160
50%	0.625 ± 0.010	0.660 ± 0.009	0.681	5160	0.514 ± 0.005	0.554 ± 0.005	0.692	5160
25%	0.629 ± 0.012	0.663 ± 0.012	0.682	5160	0.511 ± 0.005	0.550 ± 0.007	0.697	5160
12.5%	0.625 ± 0.006	0.659 ± 0.007	0.682	5160	0.510 ± 0.009	0.548 ± 0.010	0.698	5160
Only One	0.579 ± 0.020	0.614 ± 0.020	0.683	5160	0.482 ± 0.022	0.520 ± 0.022	0.697	5160

T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

Appendix J.3 Scaling the Number of Queries

In dense macromolecular environments, such as specific tomograms within the EMPIAR-10304 dataset, the total number of target particles frequently exceeds the default query capacity ($M = 900$) typical of DETR-like architectures. To address this bottleneck without the computational burden of retraining, we investigate the zero-shot scaling of our query capacity by directly duplicating the learned content queries during inference. The quantitative results are detailed in Table A45.

Remarkably, scaling the number of queries to 1350 (1.5×) and 1800 (2.0×) yields steady improvements in detection performance across all prompting configurations. However, this performance gain involves a distinct computational trade-off: peak VRAM consumption exhibits a severe super-linear growth. In practice, attempting to train FullTilt from scratch with 1800 queries strictly exceeds the 48GB VRAM capacity of an NVIDIA RTX 6000 Ada GPU, resulting in out-of-memory errors. Consequently, developing more memory-efficient strategies to handle ultra-dense particle distributions remains an important direction for future work.

Table A45: **Ablation study on zero-shot query scaling (EMPIAR-10304).** We show the impact of heuristically duplicating the default learned content queries ($M = 900$) during inference to accommodate high-density particle distributions. While scaling the query capacity to 1.5× and 2.0× strictly improves detection performance across all prompting settings, it incurs a super-linear penalty in peak VRAM consumption.

	1 Prompt				2 Prompts				4 Prompts			
# Queries	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**
900 (1.0×)	0.199 ± 0.004	0.437 ± 0.002	0.229	2410	0.200 ± 0.004	0.437 ± 0.003	0.230	2410	0.201 ± 0.004	0.439 ± 0.001	0.230	2410
1350 (1.5×)	0.218 ± 0.004	0.496 ± 0.002	0.296	3950	0.219 ± 0.004	0.497 ± 0.002	0.293	3950	0.219 ± 0.002	0.498 ± 0.001	0.295	3950
1800 (2.0×)	0.224 ± 0.002	0.515 ± 0.002	0.356	6060	0.225 ± 0.005	0.516 ± 0.002	0.356	6060	0.225 ± 0.002	0.516 ± 0.003	0.356	6060

T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

Appendix J.4 Additional Component Ablation Results

We conduct a comprehensive ablation study to isolate the impact of our proposed modules: the tilt-series encoder ($\mathcal{E}_{\text{Tilt}}$), the tilt-aware query initializer ($\mathcal{I}_{\text{Tilt}}$), and the auxiliary geometric primitives module ($\mathcal{G}$). Quantitative results on the simulation datasets are detailed in Table A46. As expected, the naive baseline—omitting all proposed modules—exhibits severely degraded performance. When evaluating components in isolation, the integration of the tilt-aware query initializer ($\mathcal{I}_{\text{Tilt}}$) yields the most substantial independent performance gains. In dual-module configurations, the combination of $\mathcal{E}_{\text{Tilt}}$ and $\mathcal{I}_{\text{Tilt}}$ demonstrates a strong synergistic effect, significantly elevating detection capabilities. Finally, incorporating the auxiliary geometric primitives module ($\mathcal{G}$) completes the FullTilt framework, achieving optimal peak performance across all evaluated metrics.

Table A46: **Ablation study on FullTilt components (simulation datasets).** We show the impact of the tilt-series encoder ($\mathcal{E}_{\text{Tilt}}$), tilt-aware query initializer ($\mathcal{I}_{\text{Tilt}}$), and auxiliary geometric primitives ($\mathcal{G}$) on detection performance across two simulation datasets. The integration of all three modules yields the optimal performance.

			Simulation (Seen)				Simulation (Unseen)			
$\mathcal{E}_{\text{Tilt}}$	$\mathcal{I}_{\text{Tilt}}$	$\mathcal{G}$	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**
			0.001 ± 0.000	0.010 ± 0.001	0.319	4790	0.000 ± 0.000	0.003 ± 0.001	0.320	4790
✓			0.004 ± 0.001	0.011 ± 0.002	0.677	4810	0.004 ± 0.001	0.008 ± 0.002	0.670	4810
	✓		0.332 ± 0.006	0.545 ± 0.007	0.331	5140	0.237 ± 0.008	0.439 ± 0.018	0.331	5140
		✓	0.027 ± 0.002	0.103 ± 0.005	0.323	4790	0.012 ± 0.002	0.068 ± 0.006	0.322	4790
✓	✓		0.558 ± 0.013	0.624 ± 0.015	0.678	5160	0.455 ± 0.014	0.523 ± 0.015	0.692	5160
	✓	✓	0.382 ± 0.010	0.569 ± 0.013	0.331	5140	0.290 ± 0.014	0.460 ± 0.012	0.330	5140
✓		✓	0.014 ± 0.003	0.032 ± 0.002	0.677	4810	0.008 ± 0.002	0.018 ± 0.001	0.660	4810
✓	✓	✓	0.632 ± 0.006	0.670 ± 0.007	0.682	5160	0.502 ± 0.012	0.542 ± 0.013	0.694	5160

✓: has this feature; T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

Appendix J.5 Detailed Ablation Study of the Tilt-Series Encoder ($\mathcal{E}_{\text{Tilt}}$)

We conducted a granular ablation study to evaluate the specific components of the tilt-series encoder: the integration of tilt embeddings, the application of alternating-attention, and the adoption of global row attention. Comprehensive results for the real-world and simulation datasets are presented in Tables A47 and A48, respectively.

As anticipated from our PETR benchmark, relying solely on static tilt embeddings without cross-tilt information fusion yields minimal performance improvement. However, cross-tilt fusion intrinsically depends on these embeddings; omitting them deprives the network of the explicit angle information necessary to geometrically correlate projections. While introducing dense alternating-attention successfully enables this fusion and drives a dramatic increase in detection performance, it incurs severe computational penalties. This dense formulation increases runtime by $10\times$ to $20\times$ and demands excessive VRAM, strictly necessitating high-capacity NVIDIA H200 GPUs for training. Although integrating hardware-aware optimizations like FlashAttention [46] reduces the memory footprint to $\mathcal{O}(N)$, the temporal complexity fundamentally remains $\mathcal{O}(N^2)$. By transitioning to our proposed global row attention mechanism, we achieve an approximate $10\times$ reduction in inference runtime, while fully preserving the robust performance gains established by the cross-tilt information fusion.

Table A47: **Ablation study of the tilt-series encoder ($\mathcal{E}_{\text{Tilt}}$) on real-world datasets.** Quantitative evaluation of the components within the tilt-series encoder, including the tilt embedding, alternating-attention (AA), and global row attention is shown. Performance is measured across the DS-10440, EMPIAR-10304, and EMPIAR-10499 datasets. Incorporating global row attention achieves an order-of-magnitude reduction in inference runtime while preserving the significant performance gains established by cross-tilt information fusion.

			DS-10440				EMPIAR-10304				EMPIAR-10499			
Tilt Embedding	AA	Global Row Attn.	mAP@0.5r ↑	mAP@1r ↑	T. ↓	M. ↓	mAP@0.5r ↑	mAP@1r ↑	T. ↓	M. ↓	mAP@0.5r ↑	mAP@1r ↑	T. ↓	M. ↓
			0.094 ± 0.004	0.153 ± 0.005	0.222	3430	0.030 ± 0.001	0.181 ± 0.003	0.142	2400	0.033 ± 0.001	0.127 ± 0.003	0.207	3380
✓			0.099 ± 0.006	0.149 ± 0.008	0.224	3430	0.031 ± 0.002	0.198 ± 0.004	0.140	2400	0.048 ± 0.001	0.167 ± 0.004	0.207	3380
✓	✓		0.125 ± 0.007	0.158 ± 0.009	3.73	3430	0.176 ± 0.005	0.384 ± 0.005	0.723	2420	0.152 ± 0.003	0.280 ± 0.004	2.05	3390
	✓	✓	0.084 ± 0.004	0.134 ± 0.005	0.512	3430	0.062 ± 0.002	0.239 ± 0.002	0.231	2420	0.049 ± 0.001	0.164 ± 0.004	0.384	3390
✓	✓	✓	0.118 ± 0.013	0.148 ± 0.013	0.499	3430	0.199 ± 0.004	0.437 ± 0.002	0.229	2410	0.159 ± 0.004	0.288 ± 0.004	0.389	3410

✓: has this feature; AA: alternating-attention; Global Row Attn.: global row attention; T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

Table A48: **Ablation study of the tilt-series encoder ($\mathcal{E}_{\text{Tilt}}$) on simulation datasets.** Quantitative evaluation of the tilt-series encoder's components under both "seen" and "unseen" simulated settings is shown. Consistent with real-world evaluations, the transition to global row attention reduces computational overhead, dramatically accelerating inference with near-zero degradation in detection performance.

			Simulation (Seen)				Simulation (Unseen)			
Tilt Embedding	AA	Global Row Attn.	mAP@0.5r ↑	mAP@1r ↑	T. ↓	M. ↓	mAP@0.5r ↑	mAP@1r ↑	T. ↓	M. ↓
			0.382 ± 0.010	0.569 ± 0.013	0.331	5140	0.290 ± 0.014	0.460 ± 0.012	0.330	5140
✓			0.377 ± 0.005	0.571 ± 0.009	0.333	5140	0.283 ± 0.014	0.450 ± 0.017	0.331	5140
✓	✓		0.639 ± 0.006	0.669 ± 0.006	6.07	5160	0.497 ± 0.015	0.535 ± 0.013	5.89	5160
	✓	✓	0.511 ± 0.014	0.617 ± 0.009	0.697	5160	0.405 ± 0.008	0.514 ± 0.009	0.688	5160
✓	✓	✓	0.632 ± 0.006	0.670 ± 0.007	0.682	5160	0.502 ± 0.012	0.542 ± 0.013	0.694	5160

✓: has this feature; AA: alternating-attention; Global Row Attn.: global row attention; T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

Appendix J.6 Detailed Ablation Study of the Tilt-Aware Query Initializer ($\mathcal{I}_{\text{Tilt}}$)

We conducted a detailed ablation study to evaluate the specific initialization strategies of the tilt-aware query initializer. Specifically, we ablate the design choice of initializing the 3D anchors' x- and y-coordinates from the closest 0° projection, as well as explicitly setting the initial z-coordinate to 0.5 (representing the geometric midplane of the z-axis). Quantitative results for the real-world and simulation datasets are provided in Tables A49 and A50, respectively.

Without explicit initialization (i.e., relying entirely on randomly initialized, fully learnable 3D anchors), the model's detection performance is severely degraded. Utilizing the closest 0° projection to initialize the x- and y-coordinates of each 3D anchor provides a critical spatial prior, dramatically improving overall performance. Finally, incorporating the depth-wise z-coordinate initialization stabilizes the spatial learning process, allowing the model to achieve optimal performance. This full initialization strategy yields particularly significant gains on the EMPIAR-10304 and EMPIAR-10499 datasets.

Table A49: **Ablation study of the Tilt-Aware Query Initializer ($\mathcal{I}_{\text{Tilt}}$) on real-world datasets.** We show the impact of explicitly initializing the x-, y-, and z-coordinates of the 3D anchors prior to the decoder stages. Relying on purely learned anchors fails to establish a strong spatial prior, whereas our proposed 0° projection initialization and midplane z-initialization drastically improve detection performance.

		DS-10440				EMPIAR-10304				EMPIAR-10499			
0° XY Init.	**Midplane Z Init.**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**
		0.002 ± 0.001	0.008 ± 0.002	0.520	3430	0.001 ± 0.000	0.017 ± 0.001	0.229	2300	0.000 ± 0.000	0.004 ± 0.000	0.379	3190
✓		0.122 ± 0.008	0.152 ± 0.008	0.510	3430	0.139 ± 0.003	0.348 ± 0.004	0.229	2420	0.155 ± 0.007	0.284 ± 0.011	0.389	3390
✓	✓	0.118 ± 0.013	0.148 ± 0.013	0.499	3430	0.199 ± 0.004	0.437 ± 0.002	0.229	2410	0.159 ± 0.004	0.288 ± 0.004	0.389	3410

✓: has this feature; Init.: initialization; T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

Table A50: **Ablation study of the Tilt-Aware Query Initializer ($\mathcal{I}_{\text{Tilt}}$) on simulation datasets.** We show the impact of explicitly initializing the x-, y-, and z-coordinates of the 3D anchors prior to the decoder stages. Relying on purely learned anchors fails to establish a strong spatial prior, whereas our proposed 0° projection initialization and midplane z-initialization drastically improve detection performance.

		Simulation (Seen)				Simulation (Unseen)			
0° XY Init.	**Midplane Z Init.**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**	**mAP@0.5r ↑**	**mAP@1r ↑**	**T. ↓**	**M. ↓**
		0.014 ± 0.003	0.032 ± 0.002	0.677	4810	0.008 ± 0.002	0.018 ± 0.001	0.660	4810
✓		0.614 ± 0.013	0.650 ± 0.012	0.681	5160	0.493 ± 0.011	0.534 ± 0.010	0.691	5160
✓	✓	0.632 ± 0.006	0.670 ± 0.007	0.682	5160	0.502 ± 0.012	0.542 ± 0.013	0.694	5160

✓: has this feature; Init.: initialization; T.: runtime (s); M.: VRAM (MB); ↓: lower is better; ↑: higher is better

Algorithm A1 PyTorch style pseudocode for Local and Global Row Attention

```
def local_attention(x):
    """
    x:      features shaped (N, H, W, C)
    """
    N, H, W, C = x.shape

    # [N, H, W, C] -> [N, H×W, C]
    x = x.flatten(1, 2)
    x = SelfAttention(x)

    # [N, H×W, C] -> [N, H, W, C]
    x = x.view(N, H, W, C)

    return x


def global_row_attention(x):
    """
    x:      features shaped (N, H, W, C)
    """
    N, H, W, C = x.shape

    # [N, H, W, C] -> [H, N, W, C]
    x = x.permute(1, 0, 2, 3)

    # [H, N, W, C] -> [H, N×W, C]
    x = x.flatten(1, 2)
    x = SelfAttention(x)

    # [H, N×W, C] -> [H, N, W, C]
    x = x.view(H, N, W, C)

    # [H, N, W, C] -> [N, H, W, C]
    x = x.permute(1, 0, 2, 3)

    return x
```

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims regarding our architectural contributions, computational efficiency, and state-of-the-art zero-shot performance are clearly stated in the abstract and introduction. These claims are fully substantiated by our mathematical formulation in Section 3 and our extensive empirical evaluations in Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We explicitly discuss the limitations in Section 5, with further details provided in Appendices C and D.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: We propose an architecture for direct 2D-to-3D macromolecule detection. Theoretical proofs are not applicable to this paper.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: All architectural, data preparation, and implementation details required to fully reproduce our main experimental results are thoroughly documented in Sections 3 and 4, and expanded upon in Appendices B–G.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: In the supplementary materials, we provide confidential inference code, environment-building instructions, pretrained weights, and a sample dataset to allow reviewers to faithfully reproduce a single instance of FullTilt inference. Due to the massive scale of the full training and evaluation datasets (>2TB), these datasets, along with the complete training code, will be made publicly accessible upon the acceptance of this manuscript.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Comprehensive training and evaluation details, including hyperparameters, optimization strategies, and data splits, are explicitly provided in Sections 3.6 and 4, and further detailed in Appendices D–G.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All main results tables report the mean and standard deviation ($\pm$ 1 standard deviation) to account for variability arising from random prompt selection and initialization. These metrics are computed over 10 independent trials for most experiments, and 3 trials for weakly supervised baselines.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We explicitly detail our hardware setup in Sections 4.1 and 4.2, specifying the use of four NVIDIA 6000 Ada GPUs and a 48-core CPU for training, and a single GPU for inference. Total training time (approximately two weeks) is documented in Appendix F.2, while all inference runtimes and VRAM requirements are comprehensively reported across our results tables.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research presented in this paper focuses on computational methods for structural biology. It utilizes exclusively publicly available datasets that contain no personally identifiable information, and the proposed algorithmic contributions present no foreseeable negative societal impacts, fully complying with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: Our work focuses strictly on foundational computational methods for structural biology and visual proteomics (cryo-ET). While this framework offers significant positive impacts by accelerating biological research and potential drug discovery, it operates exclusively on molecular imaging data. Consequently, there is no direct path to foreseeable negative societal impacts, malicious dual-use applications, or privacy and fairness concerns.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The models and datasets utilized in this work are strictly restricted to the structural analysis of macromolecules in cryo-ET imaging. As our framework lacks generative capabilities, does not rely on scraped internet data, and presents no dual-use or high-risk potential, specialized release safeguards are not applicable.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All external assets utilized in this work, including the CZII CryoET Object Identification Challenge dataset and EMPIAR datasets and baseline codebases, are properly credited via formal citation. We strictly comply with their respective open-source licenses and terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The architectural design, training protocols, and implementation details for our newly proposed framework (FullTilt) are comprehensively documented in the main manuscript and appendices. Complete codebase documentation, including a structured README, will accompany the public release of the code and pretrained weights upon acceptance.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This research does not involve crowdsourcing, user studies, or human subjects. All experiments are conducted computationally using publicly available, anonymized molecular imaging datasets.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: As this research strictly involves computational methods applied to existing, anonymized molecular imaging datasets, it does not involve human subjects. Consequently, Institutional Review Board (IRB) approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not used as a component of the core methodology in this research. They were utilized strictly for manuscript editing and formatting, which did not impact the scientific rigor or originality of the work.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.