

UC San Diego

UC San Diego Electronic Theses and Dissertations

Title

Adaptive Deep Learning for Vision and Multimodal Understanding with Heterogeneous Data

Permalink

<https://escholarship.org/uc/item/3h17286t>

ISBN

9798193289784

Author

Liang, Youwei

Publication Date

2026-10-12

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA SAN DIEGO

Adaptive Deep Learning for Vision and Multimodal Understanding with Heterogeneous Data

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Electrical Engineering (Machine Learning and Data Science)

by

Youwei Liang

Committee in charge:

Professor Pengtao Xie, Chair
Professor Manmohan Chandraker
Professor Zhuowen Tu
Professor Nuno Vasconcelos
Professor Xiaolong Wang

Copyright
Youwei Liang, 2026
All rights reserved.

The Dissertation of Youwei Liang is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2026

DEDICATION

*To my parents,
Ganping Liang and Xiuying Ye,
for their unconditional love and support.*

EPIGRAPH

What I cannot create, I do not understand.

—Richard Feynman

TABLE OF CONTENTS

Dissertation Approval Page	iii
Dedication	iv
Epigraph	v
Table of Contents	vi
List of Figures	ix
List of Tables	xi
Acknowledgements	xiii
Vita	xvi
Abstract of the Dissertation	xviii
Chapter 1 Introduction	1
1.1 Motivation: Challenges of Learning with Heterogeneous Data	1
1.2 Existing Strategies for Handling Heterogeneity	5
1.3 Adaptive Learning: Selection, Alignment, and Supervision	7
1.4 Contributions	9
1.5 Outline	13
Chapter 2 EViT: Adaptive Token Selection for Efficient Vision Transformers	15
2.1 Chapter Overview	15
2.2 Motivation	15
2.3 Related Work Specific to Efficient Vision Transformers	18
2.3.1 Vision Transformers	18
2.3.2 Model acceleration	18
2.4 Method: Token Reorganization	19
2.4.1 ViT overview	20
2.4.2 Attentive token identification	21
2.4.3 Inattentive token fusion	22
2.4.4 Analysis	23
2.5 Experiments	24
2.6 Discussion	29
2.7 Additional Visualization	29
2.8 Extended Experiments and Ablations	29
2.9 From Adaptive Computation to Adaptive Prediction	35
Chapter 3 DrugChat: Adaptive Multimodal Alignment for Molecular Properties Prediction	43

3.1	Chapter Overview	43
3.2	Motivation	43
3.3	Method	48
3.3.1	Architecture and Prompt-Conditioned Prediction	48
3.4	Training and Implementation Details	50
3.4.1	Data collection and processing	50
3.4.2	Data leakage controls and overlap audit	51
3.4.3	Model architecture	53
3.4.4	Model training	56
3.4.5	Baselines	57
3.4.6	Prompts	58
3.4.7	Model evaluation	59
3.5	Experiments	62
3.5.1	Free-form predictions of drug properties	62
3.5.2	Performance on multi-task QSAR modeling	66
3.5.3	Held-out-task generalization on FS-Mol	71
3.5.4	Impact of large-scale pretraining	72
3.5.5	Advantages of multi-modal integration	73
3.5.6	Interactive exploration of drug properties	74
3.6	Discussion	75
3.7	Data and Code Availability	79
3.7.1	Code Availability	79
3.8	Additional Dataset Diagnostics	79
3.9	From Adaptive Prediction to Adaptive Supervision	80
Chapter 4	RobPicker: Adaptive Supervision for Cryo-ET Particle Picking	81
4.1	Chapter Overview	81
4.2	Motivation	81
4.3	Method Overview	86
4.3.1	RobPicker overview	86
4.4	Bi-Level Optimization	87
4.4.1	The RobPicker framework	87
4.4.2	Optimization algorithm	89
4.5	Experiments	91
4.5.1	RobPicker tolerates noise and class imbalance in cryo-ET particle picking	92
4.5.2	RobPicker enables generalizable particle picking across organisms	98
4.5.3	Data reweighting and label correction networks both contribute to RobPicker's robustness	99
4.6	Model and Experimental Details	101
4.6.1	Model architecture	101
4.6.2	Data preprocessing	104
4.6.3	Inference with a trained network	106
4.6.4	Datasets	107
4.6.5	Experimental settings	109

4.6.6	Baseline settings	111
4.7	Discussion	113
4.8	Data and Code Availability	117
4.8.1	Code Availability	117
4.9	Toward General Adaptive Learning Systems	117
Chapter 5	Conclusions and Future Directions	119
5.1	Conclusions	119
5.2	Limitations	120
5.3	Future Directions	122
5.4	Closing Perspective	123
Bibliography		124

LIST OF FIGURES

Figure 2.1.	DeiT-S (Touvron et al., 2021a) predictions with incomplete input image tokens. In (a), removing image tokens unrelated to the visual content of the corresponding category does not deteriorate ViT predictions. In (b), removing related image tokens makes ViT predict incorrectly.....	17
Figure 2.2.	Token reorganization within a single Transformer encoder.	22
Figure 2.3.	Visualization of inattentive tokens on EViT-DeiT-S with 12 layers.....	24
Figure 2.4.	Comparison of different models with various accuracy-throughput trade-off.	28
Figure 2.5.	Extended visualizations of token reorganization in 12-layer EViT-DeiT-S. Unmasked regions are attentive tokens; masked regions are inattentive tokens fused into a new token. The examples span diverse ImageNet categories.	38
Figure 2.6.	The box plot of the evolution of the attention scores of the inattentive tokens with the layer depth. The orange line in the box plot indicates the averaged attention scores of the inattentive tokens at a certain layer.	41
Figure 2.7.	Evolution of token reorganization across EViT training epochs and layers. Unmasked regions are attentive tokens; masked regions are inattentive tokens fused into a new token. Early masks are unstable but converge to meaningful regions as training proceeds.	42
Figure 3.1.	DrugChat is a multi-modal LLM capable of predicting drug properties in either free-form texts or as discrete categories.	47
Figure 3.2.	DrugChat outperformed GPT-4 and Galactica in predicting drug indications, pharmacodynamics, mechanisms of action, and overviews expressed in free-form texts.	64
Figure 3.3.	Examples of free-form predictions generated by DrugChat, fine-tuned Galactica, and GPT-4, with their human-evaluation scores.	65
Figure 3.4.	Cytotoxicity and ChEMBL structured-task evaluation.	69
Figure 3.5.	MoleculeNet evaluation.	70
Figure 3.6.	Held-out-task generalization on FS-Mol.	71
Figure 3.7.	Ablation analysis of pretraining and molecular encoders.	72

Figure 3.8.	An example multi-turn dialogue about ramipril. DrugChat correctly identified the indication and mechanism of ramipril, while maintaining the conversation context across turns.	74
Figure 3.9.	Human-scored multi-turn evaluation over 50 compounds and 148 follow-up turns.	75
Figure 3.10.	Our curated dataset features a diverse distribution of compound categories. The inner disk represents the compound superclasses, while the outer ring shows the corresponding subclasses.	79
Figure 3.11.	The distribution of ground-truth answers in the test set across the tasks of predicting cytotoxicity, administration routes, and prodrug status.	80
Figure 4.1.	Overview of RobPicker.	85
Figure 4.2.	RobPicker performance across real cryo-ET datasets.	92
Figure 4.3.	RobPicker shows high accuracy and strong cross-organism generalizability in particle picking.	97
Figure 4.4.	RobPicker significantly outperformed DeepFinder on the SHREC 2019 dataset.	100

LIST OF TABLES

Table 2.1.	ImageNet classification accuracy of a straightforward inattentive token removal for a trained DeiT-S (Touvron et al., 2021a). The inattentive tokens are directly removed based on the attention from the class token to other tokens at the 4 th , 7 th and 10 th layers.	20
Table 2.2.	Comparison on the two variants of EViT on DeiT-S (Touvron et al., 2021a).	25
Table 2.3.	Results of EViT on LV-ViT-S	25
Table 2.4.	Results of training EViT-DeiT-S with a keep rate of 0.7 for different epochs.	25
Table 2.5.	Results of training/finetuning on high resolution images.	36
Table 2.6.	Results of training EViT-DeiT-S and EViT-DeiT-B using DeiT-S as an oracle. Training for longer epochs continues to benefits the EViT in efficiency regime.	36
Table 2.7.	Comparisons on EViT and DynamicViT (Rao et al., 2021). ✓ indicates the model is initialized with a pre-trained DeiT-S. For fair comparison, the throughput is measured on the same machine with the same setting using a maximum batch size.	37
Table 2.8.	Results of EViT on DeiT-B	37
Table 2.9.	Comparisons of the two variants (i.e., vanilla inattentive token removal and inattentive token fusion) of EViT-DeiT-S on higher resolution images. . . .	37
Table 2.10.	The performance of EViT-DeiT-S with different attentive token identification strategies.	39
Table 2.11.	Training time for 300-epoch training of DeiT-S and EViT-DeiT-S with different keep rates.	39
Table 2.12.	Results of training/finetuning on high resolution images. The numbers before and after ↑ indicate the image size in the 300-epoch training and 50-epoch finetuning, respectively.	39
Table 2.13.	The performances of EViT-DeiT-S with different of keep rates and reorganization locations (R.L.). For fair comparison, we keep the variants having the same level of computational cost (i.e., MACs) as our EViT baseline in the first row by tuning the keep rates.	40

Table 2.14.	The performance of EViT-DeiT-S equipped with the DINO attention. The “MACs” in the table is the computational cost of the EViT-DeiT-S model and “Combined MACs” is the sum of the MACs of both EViT-DeiT-S and the DINO model (i.e., a DeiT-S) which is used to obtain the DINO attention.	40
Table 2.15.	An ablation of token removal strategy in EViT-DeiT-S.	40
Table 3.1.	Residual close analogs after exact-structure and standardized-variant exclusion.....	52

ACKNOWLEDGEMENTS

I am deeply grateful to Professor Pengtao Xie for his support and guidance throughout my PhD journey. His thoughtful feedback has profoundly influenced my development as a researcher, helping to refine my academic writing and strengthen my presentation skills. I am also sincerely grateful to Professors Manmohan Chandraker, Zhuowen Tu, Nuno Vasconcelos, and Xiaolong Wang for serving on my thesis committee. I greatly appreciate the time they devoted to my work, as well as their support, insightful comments, and valuable feedback throughout this process.

I am fortunate to have worked alongside wonderful labmates, including Ramtin Hosseini, Sai Ashish Somayajula, Li Zhang, Han Guo, Ruiyi Zhang, Qi Cao, and Peijia Qin. They have been not only excellent research collaborators but also genuine friends who have supported and helped me both inside and outside the lab.

I am also grateful to Chongjian Ge, Zhan Tong, Hamidreza Rahmani, Digvijay Singh, Peizhao Li, Kevin Stone, and many other collaborators outside the lab. Their contributions have played an important role in my research, and working with them has broadened my perspective and positively influenced the way I think about and conduct research.

I would also like to express my gratitude to my industry mentors, including Yibing Song, Yuandong Tian, Junfeng He, Gang Li, and Caleb Ellington, for their guidance and support during my internships. Working with them provided me with invaluable experience, exposed me to new perspectives on research, and led to many rewarding research outcomes.

I am deeply thankful to Chengyang Huang, Kexin Weng, Xiaohan Fu, Xu Shang, Guofeng Qian, Cindy Xiao, He Gao, Xihe Gu, Ruisi Zhang, Mingjia Huo, Shuheng Li, Junhao Qu, Haiwei Lin, Jeffrey Li, Tian Liang, Yida Zou, Zhenzhe He, Zhenghua Ye, Zhaoyi Liu, Chen Huang, Yingjian Huang, Xiao Yang, Yuxin Zhang, Junhao Liu, Yuying Ge, Maoyan Chen, Mingdeng Cao, Shoufa Chen, Yuanfeng Ji, Zixuan Ke, and many other friends, including those mentioned above, who shared this journey with me. Some of my fondest memories from these years are of the time we spent playing sports, enjoying good food, and exploring San Diego together. Beyond these memories, I have learned many admirable qualities from them, and their friendship has

influenced the way I think, interact with others, and grow as a person.

I would like to express my deepest gratitude to Professor Dong Huang, who guided me at the beginning of my academic journey during my undergraduate studies. He taught me how to conduct research, read the literature critically, and write academic papers. He gave me the confidence to read difficult papers and pursue research problems I was not familiar with. Through his guidance, I gradually grew into a more independent researcher, capable of pursuing novel research ideas on my own.

I am profoundly grateful to my parents, Ganping Liang and Xiuying Ye, for raising me and for everything they have done to make my education possible. Through their diligence and dedication to their work, they gave me the opportunity to pursue higher education, and through their example, they taught me the importance of hard work and honesty. They have always been role models for me. Their open-mindedness gave me the freedom to follow an unconventional educational path, one that ultimately led me to pursue a PhD. Their unwavering care and love have been a constant source of strength throughout my life. I am also deeply grateful to my brother, Youjia Liang, and my sister, Wenxin Liang, who have been not only my family but also my companions as I grew up. The hundreds of books they gave me played a formative role in my early intellectual development, nurturing my love of reading and my curiosity about scientific questions. My family has supported and cared for me in every way, and I cannot adequately express how grateful I am to them.

I want to thank the city of San Diego. I have laughed, cried, struggled, failed, and succeeded here. Through all of these moments, San Diego has always seemed generous and comforting, offering sunny days, starry nights, beautiful sunsets, and an endless stretch of beaches and coastline. The shores of La Jolla and the vastness of the ocean have brought me comfort through moments of weariness and sorrow. Over the years, I have been deeply influenced by this city—the resilience and openness of the sea, the optimism of its sunshine, and the warmth and kindness of the people of San Diego.

I would also like to acknowledge the National Science Foundation (NSF) and the National

Institutes of Health (NIH) for their generous grant support, which made this research possible.

I am also grateful to my co-authors who kindly approved the inclusion of the following works in this dissertation.

Chapter 2, in full, is a reprint of the material as it appears in “Not All Patches are What You Need: Expediting Vision Transformers via Token Reorganizations” by Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie, which was published in International Conference on Learning Representations as a Spotlight paper in 2022. The dissertation author was the primary researcher and author of this paper.

Chapter 3, in full, has been submitted for publication of the material as it appears in “A Multi-Modal Large Language Model for Free-Form, Open-Ended, and Interactive Prediction of Properties and Mechanisms of Candidate Drug Molecules” by Youwei Liang, Ruiyi Zhang, Zinnia Ma, Digvijay Singh, Yongce Li, Mingjia Huo, Chengzhan Gao, Hamidreza Rahmani, Satvik Bandi, Li Zhang, Robert Weinreb, Atul Malhotra, Danielle A. Grotjahn, Linda Awdishu, Trey Ideker, Michael Gilson, and Pengtao Xie. The dissertation author was the primary researcher and author of this paper.

Chapter 4, in full, has been submitted for publication of the material as it appears in “RobPicker: A Meta Learning Framework for Robust Identification of Macromolecules in Cryo-Electron Tomograms” by Youwei Liang, Ramtin Hosseini, Hamidreza Rahmani, Digvijay Singh, Ruiyi Zhang, Li Zhang, Han Guo, Sang Choe, Joan Lee, Min Xu, Eran Segal, James Zou, James Williamson, Elizabeth Villa, Danielle Grotjahn, and Pengtao Xie. The dissertation author was the primary researcher and author of this paper.

VITA

- 2020 Bachelor of Science in Information and Computing Science
South China Agricultural University, Guangzhou, China
- 2024 Master of Science in Electrical Engineering (Machine Learning and Data Science)
University of California San Diego, La Jolla, U.S.A.
- 2026 Doctor of Philosophy in Electrical Engineering (Machine Learning and Data Science)
University of California San Diego, La Jolla, U.S.A.

PUBLICATIONS

* Co-first authors.

Youwei Liang*, Junfeng He*, Gang Li*, Peizhao Li, Arseniy Klimovskiy, Nicholas Carolan, Jiao Sun, Jordi Pont-Tuset, Sarah Young, Feng Yang, Junjie Ke, Krishnamurthy Dj Dvijotham, Katherine M. Collins, Yiwen Luo, Yang Li, Kai J Kohlhoff, Deepak Ramachandran, and Vidhya Navalpakkam, “Rich Human Feedback for Text-to-Image Generation,” *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 2024.

Peizhao Li*, Junfeng He*, Gang Li*, Rachit Bhargava, Shaolei Shen, Nachiappan Valliappan, **Youwei Liang**, Hongxiang Gu, Venky Ramachandran, Golnaz Farhadi, Yang Li, Kai J Kohlhoff, and Vidhya Navalpakkam, “UniAR: A Unified model for predicting human Attention and Responses on visual content,” *Neural Information Processing Systems, NeurIPS*, 2024.

Vahid Mohammadzadeh*, **Youwei Liang***, Sasan Moghimi, Pengtao Xie, Takashi Nishida, Golnoush Mahmoudinezhad, Medi Eslani, Evan Walker, Alireza Kamalipour, Eleonora Micheletti, Jo-Hsuan Wu, Mark Christopher, Linda M Zangwill, Tara Javidi, and Robert N Weinreb, “Detection of glaucoma progression on longitudinal series of en-face macular optical coherence tomography angiography images with a deep learning model,” *British Journal of Ophthalmology, BJO*, 2024.

Li Zhang, **Youwei Liang**, Ruiyi Zhang, Amirhosein Javadi, and Pengtao Xie, “BLO-SAM: Bi-level Optimization Based Finetuning of the Segment Anything Model for Overfitting-Preventing Semantic Segmentation,” *International Conference on Machine Learning, ICML*, 2024.

Mingjia Huo*, Sai Ashish Somayajula*, **Youwei Liang**, Ruisi Zhang, Farinaz Koushanfar, and Pengtao Xie, “Token-Specific Watermarking with Enhanced Detectability and Semantic Coherence for Large Language Models,” *International Conference on Machine Learning, ICML*, 2024.

Sai Ashish Somayajula, **Youwei Liang**, Li Zhang, Abhishek Singh, and Pengtao Xie, “Generalizable and Stable Finetuning of Pretrained Language Models on Low-Resource Texts,” *Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL, 2024.

Sai Ashish Somayajula, Onkar Litake, **Youwei Liang**, Ramtin Hosseini, Shamim Nemati, David O. Wilson, Robert N. Weinreb, Atul Malhotra, and Pengtao Xie, “Improving long COVID-related text classification: a novel end-to-end domain-adaptive paraphrasing framework,” *Scientific Reports*, Sci Rep, 2024.

Youwei Liang*, Kevin Stone*, Ali Shameli, Chris Cummins, Mostafa Elhoushi, Jiadong Guo, Benoit Steiner, Xiaomeng Yang, Pengtao Xie, Hugh Leather, and Yuandong Tian, “Learning Compiler Pass Orders using Coreset and Normalized Value Prediction,” *International Conference on Machine Learning*, ICML, 2023.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie, “Not All Patches are What You Need: Expediting Vision Transformers via Token Reorganizations,” *International Conference on Learning Representations*, ICLR, 2022.

Youwei Liang, Dong Huang, Chang-Dong Wang, and Philip S. Yu, “Multi-view Graph Learning by Joint Modeling of Consistency and Inconsistency,” *IEEE Transactions on Neural Networks and Learning Systems*, TNNLS, 2022.

Chongjian Ge, **Youwei Liang**, Yibing Song, Jianbo Jiao, Jue Wang, and Ping Luo, “Revitalizing CNN Attentions via Transformers in Self-Supervised Visual Representation Learning,” *Advances in Neural Information Processing Systems*, NeurIPS, 2021.

Youwei Liang and Dong Huang, “Large Norms of CNN Layers Do Not Hurt Adversarial Robustness,” *AAAI Conference on Artificial Intelligence*, AAAI, 2021.

Youwei Liang, Dong Huang, and Chang-Dong Wang, “Consistency Meets Inconsistency: A Unified Graph Learning Framework for Multi-view Clustering,” *IEEE International Conference on Data Mining*, ICDM, 2019.

Youwei Liang*, Ramtin Hosseini*, Hamidreza Rahmani, Digvijay Singh, Ruiyi Zhang, Li Zhang, Han Guo, Sang Choe, Joan Lee, Min Xu, Eran Segal, James Zou, James Williamson, Elizabeth Villa, Danielle Grotjahn, and Pengtao Xie, “RobPicker: A Meta Learning Framework for Robust Identification of Macromolecules in Cryo-Electron Tomograms.”

Youwei Liang, Ruiyi Zhang, Zinnia Ma, Digvijay Singh, Yongce Li, Mingjia Huo, Chengzhan Gao, Hamidreza Rahmani, Satvik Bandi, Li Zhang, Robert Weinreb, Atul Malhotra, Danielle A. Grotjahn, Linda Awdishu, Trey Ideker, Michael Gilson, and Pengtao Xie, “A Multi-Modal Large Language Model for Free-Form, Open-Ended, and Interactive Prediction of Properties and Mechanisms of Candidate Drug Molecules.”

ABSTRACT OF THE DISSERTATION

Adaptive Deep Learning for Vision and Multimodal Understanding with Heterogeneous Data

by

Youwei Liang

Doctor of Philosophy in Electrical Engineering (Machine Learning and Data Science)

University of California San Diego, 2026

Professor Pengtao Xie, Chair

Large-scale pretraining has produced powerful deep learning representations, but real-world data rarely match the clean, uniform assumptions of standard transfer learning. This dissertation uses *heterogeneous data* to describe settings in which information differs in modality, representation, relevance, quality, or frequency across inputs, outputs, tasks, or supervision. Examples include molecules represented as graphs, images, strings, and text; images containing both informative and redundant regions; noisy or incomplete labels; and rare but important classes. Applying pretrained models in these settings requires deciding which information to process, how to connect different forms of information, and how much to trust each supervision signal. This dissertation develops *adaptive learning* as a unifying approach through three paradigms: adaptive

selection focuses computation on useful input components; adaptive alignment maps heterogeneous modalities, tasks, and outputs into a shared predictive space; and adaptive supervision adjusts imperfect or imbalanced training signals.

The first contribution, EViT, demonstrates adaptive selection for efficient Vision Transformers. Standard Vision Transformers devote equal computation to all image patches despite substantial redundancy. EViT uses class-token attention to identify important tokens, preserves them in subsequent layers, and fuses inattentive tokens into a compact representation rather than discarding their context. This progressive token reorganization reduces computation while maintaining accurate visual recognition.

The second contribution, DrugChat, demonstrates adaptive alignment for multimodal molecular prediction. Molecules can be represented as chemical graphs, rendered images, SMILES strings, and natural-language prompts, while predictions may require labels, probabilities, categorical decisions, or free-form descriptions. DrugChat encodes molecular graphs and images with pretrained encoders, projects them into the latent space of an instruction-following language model, and conditions generation on molecular structure and task prompts. It thereby aligns heterogeneous inputs in a shared representation space and expresses diverse prediction targets through a common language-generation interface.

The third contribution, RobPicker, demonstrates adaptive supervision for cryo-electron tomography particle picking, where annotations are expensive, incomplete, noisy, and highly imbalanced across macromolecular species. Its meta-learning framework jointly trains a segmentation network, a label correction network, and a data reweighting network. Guided by cleaner validation data, these components refine imperfect labels and adapt each subtomogram’s contribution to the loss, giving greater influence to rare or small particle classes. Together, EViT, DrugChat, and RobPicker apply adaptation at complementary levels of the learning pipeline: selecting what data to process, aligning heterogeneous representations and outputs, and learning from imperfect supervision.

Chapter 1

Introduction

1.1 Motivation: Challenges of Learning with Heterogeneous Data

Modern artificial intelligence has increasingly been shaped by a pretraining paradigm. Instead of training a separate model from scratch for every task, high-capacity neural networks are first trained on large and broad data corpora, then adapted to downstream applications. This shift has been especially visible after the Transformer architecture (Vaswani et al., 2017a), whose sequence modeling machinery has supported large-scale representation learning in language, vision, multimodal reasoning, and scientific domains. Notable examples include language models such as BERT (Devlin et al., 2018), GPT-3 (Brown et al., 2020), GPT-4 (Achiam et al., 2023), and LLaMA (Touvron et al., 2023); vision models such as Vision Transformers (Dosovitskiy et al., 2021), masked autoencoders (He et al., 2022), and Segment Anything (Kirillov et al., 2023); vision-language models such as CLIP (Radford et al., 2021), BLIP and BLIP-2 (Li et al., 2022, 2023), LLaVA (Liu et al., 2024b), and MiniGPT-4 (Zhu et al., 2024); and pretrained molecular models based on graphs, images, and language-like molecular strings (Hu et al., 2020b; Zeng et al., 2022a; Zhang et al., 2021).

The success of these models suggests a powerful recipe: learn general-purpose representations from abundant data, then transfer those representations to more specialized tasks. This recipe is attractive because real applications often have limited task-specific labels, while large

unlabeled or weakly labeled corpora are easier to collect. A pretrained model can provide visual features, linguistic knowledge, molecular priors, or multimodal alignment that would be difficult to learn from a small downstream dataset alone. In this sense, pretraining has become a central infrastructure for modern AI, much like a shared computational substrate that many tasks can reuse.

However, this dissertation starts from the observation that simple transfer of a pretrained model is often not enough. Real-world tasks rarely match the clean assumptions under which a model was pretrained or fine-tuned. The data encountered in deployment may differ in modality, representation, relevance, quality, and frequency. A model may receive several forms of input rather than one, may need to produce several kinds of output rather than a fixed class label, may process examples that contain large amounts of irrelevant information, and may learn from supervision that is noisy or highly imbalanced. I use the term *heterogeneous data* to refer to this broader family of mismatches: the information available for learning is not uniform in form, structure, quality, scale, or usefulness. Heterogeneity can appear in the input, in the output, in the task specification, and in the supervision signal.

Multimodality Multimodality is one of the most visible forms of heterogeneous data. A single-modal pretrained model can learn a powerful representation of images, text, graphs, or sequences, but this representation is not automatically sufficient when a real task requires multiple modalities at once. In drug discovery, for example, the same molecule may be represented as an atom-bond graph, a two-dimensional molecular image, a SMILES string, a set of numerical properties, and textual descriptions of biological activity. In computer vision, an image may be paired with captions, bounding boxes, segmentation masks, metadata, or natural-language instructions. In structural biology, cryo-electron tomograms may be associated with particle coordinates, subtomogram averages, segmentation maps, and acquisition conditions. Each modality captures a different aspect of the underlying object or experiment, and these aspects are not trivially interchangeable.

This heterogeneity appears not only on the input side but also on the output side. Some tasks require discrete decisions such as class labels, active-or-inactive predictions, or the presence of a particle. Others require continuous values such as binding affinity, toxicity score, or molecular property measurements. Still others require spatial outputs such as boxes, masks, or coordinates, or open-ended natural-language outputs such as explanations, rationales, and summaries. A pretrained single-modal model with a fixed prediction head is therefore difficult to transfer directly to many real tasks. The model must learn how to align modalities, decide which representation is useful for a particular prompt or target, and map a shared representation into heterogeneous output spaces. Without this adaptation, fine-tuning risks reducing a rich real-world problem to an overly narrow input-output format.

Data redundancy Even within a single modality, data are often heterogeneous in usefulness. An image is not a uniform collection of equally informative patches: foreground regions may determine the label, background regions may be weakly relevant, and some regions may be distracting. A tomographic volume may contain a small number of target particles surrounded by empty space, unrelated cellular structures, reconstruction artifacts, and repeated texture. A molecular representation may include substructures that are central to a property of interest as well as components that are irrelevant for a given assay. This is a form of heterogeneity because different parts of the same input differ in task relevance, even though they share the same modality and data type.

Data redundancy affects both efficiency and accuracy. When a model processes all input components uniformly, it spends computation on regions or tokens that contribute little to the prediction. In architectures such as Vision Transformers, every retained token can participate in attention and feed-forward computation, so redundant tokens increase cost throughout the network. Redundant or distracting content can also affect performance by drawing attention away from informative evidence, strengthening spurious correlations, or diluting gradients during learning. A robust model should therefore not merely accept a fixed input representation, but

should adaptively decide which parts of the input deserve detailed processing and which can be compressed, skipped, or treated as context.

Noise Noise is another central form of heterogeneity because observations and labels differ in reliability. Input noise can arise from sensor limitations, compression, occlusion, motion, reconstruction artifacts, missing data, or domain-specific acquisition effects. In cryo-electron tomography, for instance, tomograms often contain low signal-to-noise ratios and missing-wedge artifacts; in molecular assays, measured properties may vary across experimental protocols; in natural images and text, irrelevant background, ambiguous wording, or corrupted examples may enter the training set. A pretrained representation can help, but it cannot by itself guarantee that downstream learning will separate signal from artifact.

Noise also appears in the output or supervision. Label noise may include wrong class labels, inaccurate particle coordinates, missing positives, false positives, uncertain boundaries, or inconsistent annotations across experts and datasets. This is especially damaging because supervised learning treats labels as the target signal to imitate. If all labels are trusted equally, a high-capacity model may fit annotation errors, propagate systematic biases, and become overconfident on unreliable examples. The challenge is therefore not simply to learn from data, but to learn while estimating which data and labels should be trusted, corrected, or down-weighted.

Data imbalance Finally, real-world data are usually imbalanced because the world itself is unevenly distributed. Common objects, frequent molecular activities, abundant particle species, majority patient groups, and easy examples can dominate the empirical training distribution, while rare but important cases may appear only sparsely. This imbalance is not only a statistical inconvenience; it is a form of heterogeneity in frequency and importance. Some examples provide abundant repeated evidence, whereas others represent rare events that may be scientifically or clinically central.

Imbalance can make models appear strong while hiding serious failures. A model op-

timized on the average loss can achieve high overall accuracy by performing well on frequent classes and poorly on rare ones. In molecular prediction, inactive compounds may greatly outnumber active compounds for many assays. In cryo-electron tomography, large or abundant macromolecular complexes may dominate training signals, while small or rare complexes are harder to detect. In vision datasets, frequent backgrounds and object categories can shape the learned representation more strongly than rare cases. Addressing imbalance requires more than adding data whenever possible; it requires learning strategies that prevent dominant examples from overwhelming the model and that preserve sensitivity to rare, difficult, or high-value cases.

These forms of heterogeneity motivate the central direction of this dissertation. If real data differ in modality, relevance, reliability, and frequency, then effective deep learning systems should not process all information in the same way. They should adaptively decide what to compute over, how to combine and condition on heterogeneous representations, and how much to trust each supervision signal. The chapters that follow study this principle at three levels of the learning pipeline: adaptive input processing in EViT, adaptive multimodal prediction in DrugChat, and adaptive supervision in RobPicker.

1.2 Existing Strategies for Handling Heterogeneity

The challenges above can be addressed from both the data side and the modeling side. In practice, a significant amount of effort is often spent before modeling begins: experts manually inspect annotations, remove corrupted samples, standardize data formats, deduplicate near-identical records, denoise measurements, augment underrepresented conditions, and rebalance training sets. These interventions can be highly effective, especially in scientific settings where domain knowledge is essential for separating meaningful signal from artifact. Related methodological tools include label-error detection and dataset auditing through confident learning (Northcutt et al., 2021), denoising and restoration methods for low-quality imaging data (Bepler et al., 2020), data augmentation methods such as mixup (Zhang et al., 2018), and class-aware sampling or

loss design for imbalanced datasets (Lin et al., 2017; Cui et al., 2019). Such data-centric work is indispensable, but it is often labor-intensive, domain-specific, and incomplete when heterogeneity remains present at training or deployment time.

On the modeling side, one broad response to multimodality is to learn shared or aligned representation spaces across modalities. Vision-language models such as CLIP align images and text through contrastive learning (Radford et al., 2021), while BLIP and BLIP-2 combine image encoders with language modeling objectives for both understanding and generation (Li et al., 2022, 2023). More recent multimodal large language models, including LLaVA and MiniGPT-4, connect visual encoders to instruction-following language models so that the same model can respond to flexible natural-language prompts (Liu et al., 2024b; Zhu et al., 2024). These works show that heterogeneous inputs and outputs can be handled by learning interfaces among modality-specific encoders, shared latent spaces, and language-conditioned prediction mechanisms.

Another line of modeling work addresses redundancy by reducing unnecessary computation or suppressing uninformative input components. Network pruning removes parameters or channels that contribute little to inference (Molchanov et al., 2016); sequence and Transformer methods filter redundant tokens or adapt computation depth (Dai et al., 2020; Liu et al., 2020); and recent Vision Transformer methods dynamically select, prune, or aggregate visual tokens according to input content (Rao et al., 2021; Wang et al., 2021b; Ryoo et al., 2021). These approaches recognize that not all parts of an input deserve equal computational budget.

For noisy supervision, the literature includes loss correction, sample selection, label correction, distillation, and meta-learning based reweighting. Representative examples include learning from noisy labels with distillation (Li et al., 2017), Co-teaching (Han et al., 2018), DivideMix (Li et al., 2020a), and validation-guided example reweighting or meta-weight networks (Ren et al., 2018; Shu et al., 2019). For imbalanced and long-tailed data, common strategies include focal loss (Lin et al., 2017), class-balanced losses based on the effective number of samples (Cui et al., 2019), margin-based objectives for imbalanced classification (Cao et al., 2019),

and decoupling representation learning from classifier learning (Kang et al., 2020). Together, these methods show that the model itself can be designed to account for differences in reliability, frequency, and task relevance.

This dissertation builds on this broader landscape while emphasizing a unifying principle: adaptation should be part of the learning system rather than only a preprocessing step. Data curation can improve the training distribution, and specialized objectives can target individual failure modes, but real applications often require models that continue to make input-, task-, and supervision-dependent decisions. The next section organizes this perspective as adaptive learning, a framework based on selection, alignment, and supervision.

1.3 Adaptive Learning: Selection, Alignment, and Supervision

The central idea of this dissertation is *adaptive learning*: deep learning systems should change how they process, represent, and learn from data according to the structure and quality of the information available. I organize this idea around three mechanisms: learning by adaptive selection, learning by adaptive alignment, and learning by adaptive supervision. Adaptive selection decides which parts of the input deserve computation. Adaptive alignment expresses heterogeneous modalities, tasks, and outputs in a shared representational space. Adaptive supervision adjusts unreliable supervision during training so that the model is guided by cleaner and better weighted learning signals.

Learning by adaptive selection Learning by adaptive selection aims to reduce unnecessary or redundant data and focus computation on the most important data components. In many settings, the input contains a mixture of informative and weakly informative regions, tokens, patches, examples, or modalities. A selection mechanism therefore asks: which parts of the data should be processed in detail, and which parts can be compressed, skipped, or treated as background context? This idea appears in dynamic inference and token-selection methods for Transformers,

including DynamicViT (Rao et al., 2021), adaptive sequence-length Vision Transformers (Wang et al., 2021b), and TokenLearner (Ryoo et al., 2021). Chapter 2 follows this direction through EViT, which uses class-token attention to preserve attentive image tokens and fuse inattentive tokens, reducing redundant computation while keeping useful contextual information.

Learning by adaptive alignment Learning by adaptive alignment addresses heterogeneity across modalities, representations, tasks, and output forms. Instead of assuming that all data can be processed by a single fixed encoder or prediction head, alignment methods learn mappings that place different modalities into compatible spaces. In vision-language learning, CLIP aligns images and text through contrastive supervision (Radford et al., 2021), while BLIP and BLIP-2 connect visual encoders with language understanding and generation objectives (Li et al., 2022, 2023). Multimodal instruction-following systems such as LLaVA and MiniGPT-4 further show how aligned representations can support flexible natural-language interaction (Liu et al., 2024b; Zhu et al., 2024). Chapter 3 applies this principle to molecular prediction: DrugChat aligns molecular graphs, molecular images, and language prompts so that heterogeneous molecular information can support both structured property prediction and open-ended textual responses.

Learning by adaptive supervision Learning by adaptive supervision focuses on the supervision signal itself. When labels are noisy, incomplete, imbalanced, or inconsistent, the model should not treat every target as equally trustworthy. Adaptive supervision methods estimate which labels or examples are reliable, correct noisy targets, or reweight training examples during optimization. Representative approaches include Co-teaching, which trains networks to select small-loss examples for each other (Han et al., 2018); DivideMix, which treats noisy-label learning as semi-supervised learning (Li et al., 2020a); and meta-learning approaches that learn example weights or soft label corrections from validation feedback (Ren et al., 2018; Shu et al., 2019; Wu et al., 2021; Zheng et al., 2021). Chapter 4 develops this idea for cryo-electron tomography by correcting noisy particle labels and reweighting examples under severe class imbalance.

Together, these three mechanisms define the dissertation’s adaptive-learning thesis: models for heterogeneous real-world data should learn what information to select, how to align diverse representations and outputs, and how to correct imperfect supervision. This view connects the three technical chapters as adaptive mechanisms operating at different levels of the learning pipeline.

1.4 Contributions

Adaptive selection for efficient visual recognition For learning by adaptive selection, this thesis proposes EViT, an adaptive token reorganization method that reduces redundancy in visual recognition with Vision Transformers. Vision Transformers represent an image as a sequence of patch tokens, but many tokens correspond to background or weakly discriminative regions. EViT starts from the observation that self-attention can operate on variable-length token sequences and that the attention from the class token provides a useful internal signal of token importance. Instead of processing every patch token uniformly throughout the network, EViT adaptively preserves the most attentive tokens and compresses the remaining inattentive tokens.

The main methodological contribution of EViT is a simple token reorganization module. At selected Transformer layers, EViT computes the attentiveness between the class token and all image tokens, keeps the top- k attentive tokens, and fuses the inattentive tokens into a single weighted token. This fusion step is important: rather than discarding all less-attended regions, EViT preserves residual contextual information and allows gradients to flow through the inattentive tokens during training. As the network becomes deeper, the token sequence is progressively shortened, reducing the cost of self-attention and feed-forward computation without adding parameters or requiring a separate token-selection network.

Concretely, EViT contributes an adaptive-selection framework for efficient ViT inference, a class-attention-based criterion for identifying informative visual tokens, an inattentive-token fusion strategy that improves stability over direct token removal, and a plug-in design that can be

incorporated into representative ViT backbones such as DeiT and LV-ViT. Empirically, EViT improves the efficiency-accuracy trade-off on ImageNet classification: for example, it speeds up DeiT-S inference by approximately 50% while reducing top-1 accuracy by only 0.3%, and it can also use the saved computation to process higher-resolution inputs, improving DeiT-S accuracy by about 1% under a similar computational budget. Visualizations and ablations further show that EViT learns to preserve class-relevant image regions while compressing less informative tokens. In the dissertation, EViT establishes the first adaptive-learning mechanism: models should learn what parts of the input to process in detail when the input contains redundant information.

Adaptive alignment for molecular prediction For learning by adaptive alignment, this thesis proposes DrugChat, a multimodal large language model for molecular property prediction. DrugChat addresses two forms of heterogeneity in molecular learning. On the input side, the same molecule can be represented as a chemical graph, a rendered molecular image, a SMILES string, and a natural-language task prompt. On the output side, molecular prediction tasks may require binary labels, continuous decision scores, categorical decisions, or free-form textual descriptions of indications, pharmacodynamics, and mechanisms of action. DrugChat aligns these heterogeneous inputs and outputs within a unified language-model interface.

The main methodological idea of DrugChat is to convert molecular representations into tokens that can be interpreted by an instruction-tuned language model. Starting from a SMILES string, DrugChat constructs both a molecular graph and a molecular image. A pretrained Graph Isomorphism Network encodes the graph structure, a pretrained ResNet encodes the molecular image, and lightweight adapters project these molecular representations into the latent space of Vicuna. The resulting molecule token is inserted into the language-token sequence formed by the user prompt, allowing the language model to generate an answer conditioned jointly on molecular structure and task specification. This design aligns different input modalities in a shared representation space and aligns different output structures through a common autoregressive text-generation format.

Concretely, DrugChat contributes a prompt-conditioned multimodal architecture for molecular property prediction, a dual-encoder representation strategy that combines graph-based and image-based representations of molecular structure, and a unified molecule-prompt-answer training formulation that covers both structured QSAR tasks and free-form drug-property descriptions. Its training combines large-scale pretraining on a decontaminated corpus of PubChem bioassay activities with finetuning on curated datasets from MoleculeNet, FS-Mol, ChEMBL, DrugBank, and cytotoxicity benchmarks. Empirically, DrugChat improves over fine-tuned Galactica and GPT-4 on human-rated free-form prediction of drug indications, pharmacodynamics, and mechanisms of action; performs competitively against GNN-based, multimodal, and language-model baselines on structured molecular prediction tasks; and generalizes to FS-Mol assays held out from downstream finetuning, including a stricter split that also excludes compounds seen in FS-Mol training tasks. The remaining overlap of protein targets and endpoint types with PubChem means that this experiment should not be interpreted as generalization to wholly novel biological tasks. In the dissertation, DrugChat establishes the second adaptive-learning mechanism: models should learn how to align heterogeneous representations, task prompts, and output formats so that molecular prediction can become flexible rather than tied to fixed task heads.

Adaptive supervision for robust cryo-ET particle picking For learning by adaptive supervision, this thesis proposes RobPicker, a meta-learning framework for robust particle picking in cryo-electron tomography. Cryo-ET particle picking is a supervision-limited problem: annotations are expensive, often incomplete or noisy, and highly imbalanced across macromolecular species. Larger and more abundant particles tend to dominate voxel-wise training losses, while small or rare particles are underrepresented and more difficult to annotate reliably. RobPicker addresses this setting by adaptively learning a cleaner and more balanced supervision signal rather than treating all training labels and examples as equally reliable.

The main methodological idea of RobPicker is to train three coupled networks: a macro-

molecule segmentation network, a label correction network, and a data reweighting network. The segmentation network predicts particle species and locations from cryo-ET subtomograms. The label correction network refines noisy particle labels, for example by recovering missing particle annotations or reducing erroneous label artifacts. The data reweighting network assigns a scalar weight to each training subtomogram or patch, controlling how strongly its loss contributes to the segmentation-network update. This weight is learned adaptively rather than fixed by a manually chosen class-frequency rule. In practice, the learned weights place greater emphasis on subtomograms containing smaller or rarer particle species, thereby increasing the loss contribution of classes that would otherwise be overwhelmed by abundant or large particles.

RobPicker learns these correction and reweighting mechanisms through validation-guided bi-level optimization. At the lower level, the segmentation network is trained on noisy training data using corrected labels and reweighted losses. At the upper level, the label correction and data reweighting networks are updated to minimize segmentation loss on a smaller validation set with cleaner labels and a more balanced species distribution. This design makes the supervision signal adaptive: labels are corrected according to their usefulness for validation performance, and loss weights are assigned according to whether examples improve generalization rather than only according to raw class frequency or prediction error.

Concretely, RobPicker contributes a unified framework for joint label correction and data reweighting in cryo-ET particle picking, a bi-level optimization algorithm for learning these auxiliary networks from cleaner validation feedback, and an instance-dependent reweighting strategy that improves sensitivity to small and rare macromolecular species. Empirically, RobPicker improves robustness across real cryo-ET datasets with noisy labels and species imbalance, outperforming established supervised methods such as DeepFinder and DeePiCt, top-ranked ensemble pipelines from a Kaggle competition for cryo-ET particle picking (Henkel and Khvedchenya, 2025; Luo and Wang, 2025; Ouertani et al., 2025), self-supervised baselines such as CryoSiam, and standard imbalance-aware losses such as class-reweighted Dice loss, Focal Loss, and Unified Focal Loss. Its gains are especially pronounced for challenging targets such as ATP synthase and

fatty acid synthase. RobPicker also improves downstream subtomogram averaging for protein structure determination, showing that adaptive supervision can support not only better detection metrics but also more reliable structural biology workflows. In the dissertation, RobPicker establishes the third adaptive-learning mechanism: models should learn how much to trust and emphasize each training signal when supervision is noisy, incomplete, and imbalanced.

1.5 Outline

Chapter 2 begins the technical core of the dissertation with EViT. This chapter studies adaptive selection in visual recognition, where a Vision Transformer must process image patches that differ substantially in their relevance to the final prediction. EViT uses class-token attention to identify informative tokens, preserves them through the transformer, and fuses less informative tokens so that computation is reduced without discarding all contextual information. The chapter frames efficient inference as a problem of adaptive input processing, showing that redundancy in visual data can be exploited to improve the tradeoff between accuracy and computation.

Chapter 3 moves from adaptive computation to adaptive prediction through DrugChat. Molecular prediction is heterogeneous on both the input side, where the same molecule can be represented as a graph, image, or SMILES sequence, and the output side, where tasks may require binary decisions, numerical estimates, categorical predictions, or free-form natural language answers. DrugChat addresses this setting by aligning molecular representations with an instruction-following language model, allowing different molecular modalities and task prompts to be expressed in a shared predictive interface. This chapter presents adaptive alignment as a way to make pretrained models usable for flexible scientific prediction tasks.

Chapter 4 then shifts from adaptive prediction to adaptive supervision through RobPicker. In cryo-electron tomography, particle-picking models must learn from annotations that are expensive, incomplete, noisy, and strongly imbalanced across particle species. RobPicker learns to correct labels and reweight training examples through validation-guided meta-learning, giving

greater influence to reliable and underrepresented supervision signals. This chapter shows that adaptivity is useful not only in inference or representation learning, but also in the training signal itself.

Chapter 5, “Conclusions and Future Directions,” closes the dissertation by synthesizing EViT, DrugChat, and RobPicker as adaptive mechanisms operating at different levels of the learning pipeline. It discusses how adaptive selection, adaptive alignment, and adaptive supervision jointly address the broader challenge of heterogeneous data, and it reflects on the remaining limitations of these approaches. The chapter closes with future directions for building efficient, robust, and flexible deep learning systems that can better adapt to the complexity of real-world vision and multimodal scientific tasks.

Chapter 2

EViT: Adaptive Token Selection for Efficient Vision Transformers

2.1 Chapter Overview

This chapter presents EViT as the dissertation’s first example of adaptive deep learning: adaptive input processing. Vision Transformers process images as sequences of patch tokens, but many tokens correspond to background or non-discriminative regions. EViT uses class-token attention to identify informative image tokens, preserves those tokens, and fuses less attentive tokens into a compact representation. In this way, the model reduces redundant computation while preserving information flow through the network.

EViT belongs in the dissertation because it establishes a general principle used by the later scientific chapters: heterogeneous inputs should not be processed uniformly when the model can adaptively identify which information is most useful.

2.2 Motivation

Computer vision research has evolved into Transformers since ViTs (Dosovitskiy et al., 2021). Equipped with global self-attention, ViTs have shown impressive capability upon local convolution (i.e., CNNs) on prevalent visual recognition scenarios, including image classification (Dosovitskiy et al., 2021; Touvron et al., 2021a; Jiang et al., 2021; Graham et al., 2021), object detection (Carion et al., 2020), and semantic segmentation (Xie et al., 2021; Liu et al., 2021;

Wang et al., 2021a,c), with both supervised and unsupervised (self-supervised) training (Pan et al., 2021; Ge et al., 2021) configurations. Based on the main spirit of ViTs (i.e., MHSA), there are wide investigations (Liu et al., 2021; Chu et al., 2021; Wang et al., 2021a) to explore the network structure of ViT models for continuous recognition performance improvement.

Along with the development of ViT models, the computation burden is becoming an issue. The global self-attention between image tokens and long-range dependency make the model converge slow compared to CNNs. As illustrated in (Dosovitskiy et al., 2021), training a ViT from scratch typically requires larger datasets (e.g., ImageNet-21k (Deng et al., 2009) and JFT-300M (Sun et al., 2017)) than those of CNNs (e.g., CIFAR-10/100 (Krizhevsky et al., 2009) and ImageNet-1k). Also, using more training iterations is a necessity for network convergence (Dosovitskiy et al., 2021; Touvron et al., 2021a; Wang et al., 2022b). Without such large scale training, the ViT models are not fully exploited and perform inferior on visual recognition scenarios. These issues motivate us to expedite ViTs for practice usage.

The model acceleration of ViTs is important to reduce computational complexity. However, there are few studies focused on ViT acceleration. This is because the significant model difference between CNNs and ViTs prevents CNN model acceleration (e.g., pruning and distillation) from applying on ViTs. Nevertheless, we analyze ViT from another perspective. We observe that not all image tokens in ViTs contribute positively to the final predictions. Fig. 2.1 shows some examples where part of the input image tokens are randomly dropped out. In Fig. 2.1b, removing image tokens related to the visual content of the corresponding category makes the ViT predict incorrectly. In comparison, removing unrelated image tokens does not affect ViT predictions, as shown in Fig. 2.1a. On the other hand, we notice that ViTs divide images into non-overlapping tokens and perform self-attention (Vaswani et al., 2017a) computation on these tokens. A notable characteristic of self-attention is that it can process a varying number of tokens. These observations motivate us to reorganize image tokens for ViT model accelerations.

In this work, we propose a token reorganization method to identify and fuse image tokens. Given all the image tokens as input, we compute token attentiveness between these tokens and the

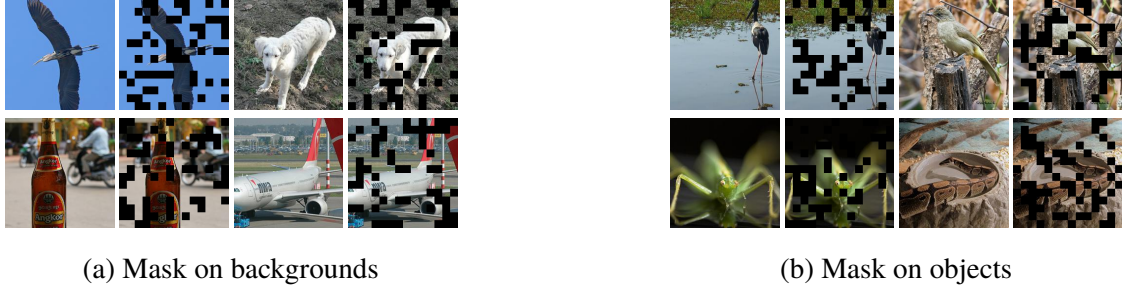


Figure 2.1: DeiT-S (Touvron et al., 2021a) predictions with incomplete input image tokens. In (a), removing image tokens unrelated to the visual content of the corresponding category does not deteriorate ViT predictions. In (b), removing related image tokens makes ViT predict incorrectly.

class token for identification. Then, we preserve the attentive image tokens and fuse the inattentive tokens into one token to allow the gradient back-propagate through the inattentive tokens for better attentive token identification. In this way, we gradually reduce the number of image tokens as the network goes deeper to decrease computation cost. Also, the capacity of the ViT backbone can be flexibly controlled via the identification process where no additional parameters are introduced. We adopt our token reorganization method on representative ViT models (i.e., DeiT (Touvron et al., 2021a) and LV-ViT (Jiang et al., 2021)) for ImageNet classification evaluation. The experimental results show our advantages. For instance, with the same amount of input image tokens, our method speeds up the DeiT-S model by 50%, while only sacrificing 0.3% recognition accuracy on ImageNet classification. On the other hand, we extend our methods to boost the ViT model recognition performance under the same computational cost. By increasing the input image resolution, our method facilitates Vision Transformers in taking more image tokens to achieve higher classification accuracy. Numerically, we improve the ImageNet classification accuracy of the DeiT-S model by 1% under the same computational cost. Moreover, by using an oracle ViT to guide the token reorganization process, our method can increase the accuracy of the original DeiT-S from 79.8% to 80.7% while reducing its computation cost by 36% under the multiply-accumulate computation (MAC) metric.

2.3 Related Work Specific to Efficient Vision Transformers

2.3.1 Vision Transformers

Transformers (Vaswani et al., 2017a) have drawn much attention to computer vision recently due to its strong capability of modeling long-range relation. A few attempts have been made to add self-attention layers or Transformers on top of CNNs in image classification (Hu et al., 2019), object detection (Carion et al., 2020), segmentation (Wang et al., 2021c), image retrieval (Lu et al., 2019) and even video understanding (Sun et al., 2019; Girdhar et al., 2019). Vision Transformer (ViT) (Dosovitskiy et al., 2021) first introduced a set of pure Transformer backbones for image classification and its follow-ups modify the ViT architecture for not only better visual recognition (Touvron et al., 2021a; Yuan et al., 2021; Zhou et al., 2021) but many other high-level vision tasks, such as object detection (Carion et al., 2020; Zhu et al., 2020; Liu et al., 2021), semantic segmentation (Wang et al., 2021a; Xie et al., 2021; Chu et al., 2021), and video understanding (Bertasius et al., 2021; Fan et al., 2021). Vision Transformers have shown its strong potential as an alternative to the previously dominant CNNs.

2.3.2 Model acceleration

Neural networks are typically overparameterized (Allen-Zhu et al., 2019), which results in significant redundancy in computation in deep learning models. To deploy deep neural networks on mobile devices, we must reduce the storage and computational overhead of the networks. Many adaptive computation methods are explored (Bengio et al., 2015; Wang et al., 2018; Graves, 2016; Hu et al., 2020a; Wang et al., 2020b; Han et al., 2021b) to alleviate the computation burden. Parameter pruning (Srinivas and Babu, 2015; Han et al., 2015; Chen et al., 2015b) reduces redundant parameters which are not sensitive to the final performance. Some other methods leverage knowledge distillation (Hinton et al., 2015; Romero et al., 2014; Luo et al., 2016; Chen et al., 2015a) to obtain a small and compact model with distilled knowledge of a larger one. These model acceleration strategies are limited to convolutional neural networks.

There are also some attempts to accelerate the computation of the Transformer model, including proposing more efficient attention mechanisms (Wang et al., 2020a; Kitaev et al., 2020; Choromanski et al., 2020) and the compressed Transformer structures (Liu et al., 2021; Heo et al., 2021; Wang et al., 2021a). These methods mainly focus on reducing the complexity of the network architecture through artificially designed modules. Another approach to ViT acceleration is reducing the number of tokens involved in the inference of ViTs. Notably, Wang et al. (2021b) proposed a method to dynamically determine the number of patches to divide on an image. The ViT will stop inference for an input image if it has sufficient confidence in the prediction of the intermediate outputs. Remarkably, Ryoo et al. (2021) proposed TokenLearner to expedite ViTs, where a relatively small amount of tokens are learned by aggregating the entire feature map weighted by a dynamic attention map conditioned on the feature map. This can be seen as a sophisticated method for tokenizing the input images. Different from TokenLearner, our work focuses on the progressive selection of informative tokens during training. Another related work is DynamicViT (Rao et al., 2021), which introduces a method to reduce token for a fully trained ViT, where an extra learnable neural network is added to ViT to select a subset of tokens. Our work provides a novel perspective for reducing the computational overhead of inference by proposing a token reorganization method to progressively reduce and reorganize image tokens. Unlike DynamicViT, our method does not need a fully trained ViT to help the training and brings no additional parameters into ViT.

2.4 Method: Token Reorganization

Our method EViT is built upon ViT (Dosovitskiy et al., 2021) and its variants for visual recognition. We first review ViT and then present how to incorporate our method into the ViT training procedure. Each component of EViT, including attentive token identification and inattentive token fusion, will be elaborated. Furthermore, we analyze the effectiveness of our method by visualizing the attentive tokens at different layers and discuss training on higher

Table 2.1: ImageNet classification accuracy of a straightforward inattentive token removal for a trained DeiT-S (Touvron et al., 2021a). The inattentive tokens are directly removed based on the attention from the class token to other tokens at the 4th, 7th and 10th layers.

Token keep rate		1.0	0.9	0.8	0.7	0.6	0.5
Top-1 Acc (%)		79.8	79.7(-0.1)	79.2(-0.6)	78.5(-1.3)	76.8(-3.0)	73.8(-6.0)

resolution images with EViT.

2.4.1 ViT overview

Vision Transformers (ViTs) are first introduced by Dosovitskiy et al. (2021) into visual recognition. They perform tokenization by dividing an input image into patches and projecting each patch to a token embedding. An extra class token [CLS] is added to the set of image tokens and is responsible for aggregating global image information and final classification. All of the tokens are added by a learnable vector (i.e., positional encoding) and fed into the sequentially-stacked Transformer encoders consisting of a multi-head self-attention (MHSA) layer and a feed-forward network (FFN). In MHSA, the tokens are linearly mapped and further packed into three matrices, namely Q , K , and V . The attention operation is conducted as follows.

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V. \quad (2.1)$$

where d is the length of the query vector. The result of $\text{Softmax}(QK^\top/\sqrt{d})$ is a square matrix which is called the attention map. The first row of attention map represents the attention from [CLS] to all tokens and will be used to determine the attentiveness (importance) of each token (detailed in the next subsection). The output tokens of MHSA are sent to FFN, consisting of two fully connected layers with a GELU activation layer (Hendrycks and Gimpel, 2016) in between. At the final Transformer encoder layer, the [CLS] token is extracted and utilized for object category prediction. More details of Transformers can be found in Vaswani et al. (2017a).

2.4.2 Attentive token identification

Let n denote the number of input tokens to a ViT encoder. In the last encoder of ViT, the [CLS] token is taken out for classification. The interactions between [CLS] and other tokens are performed via the attention mechanism (Vaswani et al., 2017a) in the ViT encoders:

$$x_{\text{class}} = \text{Softmax}\left(\frac{q_{\text{class}} \cdot K^\top}{\sqrt{d}}\right)V = a \cdot V. \quad (2.2)$$

where q_{class} , K , and V denote the query vector of [CLS], the key matrix, and the value matrix, respectively, in an attention head. In other words, the output of the [CLS] token x_{class} is a linear combination of the value vectors $V = [v_1, v_2, \dots, v_n]^\top$, with the combination coefficients (denoted by a in Eq. 2.2) being the attention values from [CLS] with respect to all tokens. Since v_i comes from the i -th token, the attention value a_i (i.e., the i -th entry in a) determines how much information of the i -th token is fused into the output of [CLS] (i.e., x_{class}) through the linear combination. It is thus natural to assume that the attention value a_i indicates the importance of the i -th token.

Moreover, Caron et al. (2021) also showed that the [CLS] token in ViTs pays more attention (i.e., having a larger attention value) to class-specific tokens than to the tokens on the non-object regions. To this end, we propose to use the attentiveness of the [CLS] token with respect to other tokens to identify the most important tokens. Based on these arguments, a simple method to reduce computation in ViT is to remove the tokens with the smallest attention values. However, we find that directly removing those tokens severely deteriorates the classification accuracy, as shown in Table 2.1. Therefore, we propose to incorporate image token reorganization during the ViT training process.

In multi-head self-attention layer, there are multiple heads performing the computation of Eq. 2.1 in parallel. Thus, there are multiple [CLS] attention vectors $a^{(h)}$, $h = [1, \dots, H]$, with H being the total number of attention heads (Vaswani et al., 2017a). We compute the average attentiveness value of all heads by $\bar{a} = \sum_{h=1}^H a^{(h)} / H$. As shown in Figure 2.2, we identify and

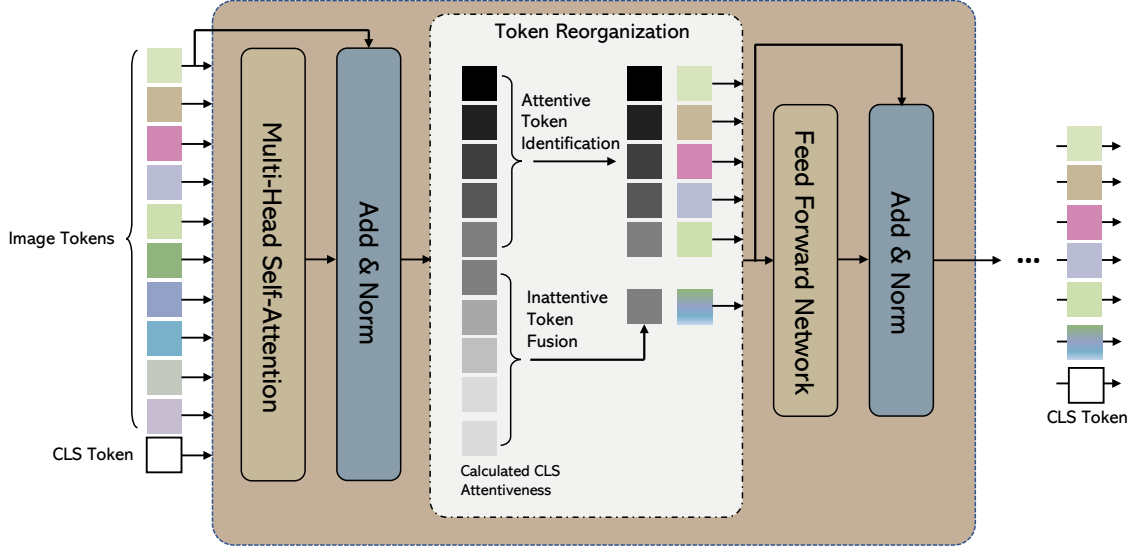


Figure 2.2: Token reorganization within a single Transformer encoder. Based on ViT (Dosovitskiy et al., 2021), we reorganize tokens in the original Transformer encoder. Specifically, we calculate the attentiveness of the class token with respect to each image token. Then, we use the attentiveness value as a criterion to identify the top- k attentive tokens and fuse the inattentive tokens.

preserve the tokens corresponding to the k largest (top- k) elements in $\bar{a}$ (k is a hyperparameter), which we call the attentive tokens, and further fuse the other tokens (which we call the inattentive tokens) into a new token. The fusion of tokens is detailed in the following paragraph. We define the token keeping rate as $\kappa = k/n$.

2.4.3 Inattentive token fusion

Although the tokens on the backgrounds of images are less informative and can be discarded without significantly influencing the performance of the ViT model, they may still be able to contribute to the prediction results. On the other hand, some images may have large object parts covering a large proportion of the images. Thus, it is possible to remove some tokens corresponding to those object parts when we select a fixed number of tokens to keep in a ViT encoder, which may negatively affect the image recognition performance. To mitigate these problems, we propose to fuse the inattentive tokens at the current stage to supplement attentive ones, as illustrated in Figure 2.2. The inattentive token fusion benefits our method to preserve

more information provided by the inattentive tokens. Specifically, we denote the indices set of the inattentive tokens as $\mathcal{N}$. The proposed inattentive token fusion is a weighted average operation, which can be written as $x_{\text{fused}} = \sum_{i \in \mathcal{N}} a_i x_i$. The fused token x_{fused} is appended to the attentive tokens and sent to the subsequent layers. The computation cost of token fusion is negligible compared to the bulk computation of ViT.

2.4.4 Analysis

Training with higher resolution images. Since our approach is efficient in processing image tokens, we are able to feed more tokens into an EViT while maintaining the computational cost at the same level as a vanilla ViT. A straightforward method to get more tokens is resizing the input images to a higher resolution and keeping the patch size unchanged. Note that these higher resolution images do not necessarily come from raw images with a higher resolution. In our vision recognition experiments on ImageNet, we simply resize the standard input images of size 224×224 to a larger spatial size (e.g., 256×256) via bicubic interpolation to obtain the higher resolution images, which are further divided into more tokens. Therefore, compared to a vanilla ViT, EViT uses *no* additional information from the images to obtain the prediction results in both training and inference. The experimental results in Table 2.5 validate the effectiveness of the proposed method.

Visualization. The proposed EViT accelerates ViTs by identifying the attentive tokens and discarding the redundant calculation on inattentive image tokens. To further investigate the interpretability of EViT, we visualize the attentive token identification procedure in Figure 2.3. We present the original images and the attentive token identification results at different layers (e.g., the 4th, 7th and 10th layer). It can be seen that as the network deepens, the inattentive tokens are gradually removed or fused, while the most informative/attentive tokens are identified and preserved. In this way, the proposed method facilitates the ViTs in focusing on class-specific tokens in images. The visualization results also validate that our EViT is effective in dealing with images with either simple backgrounds or complex backgrounds.

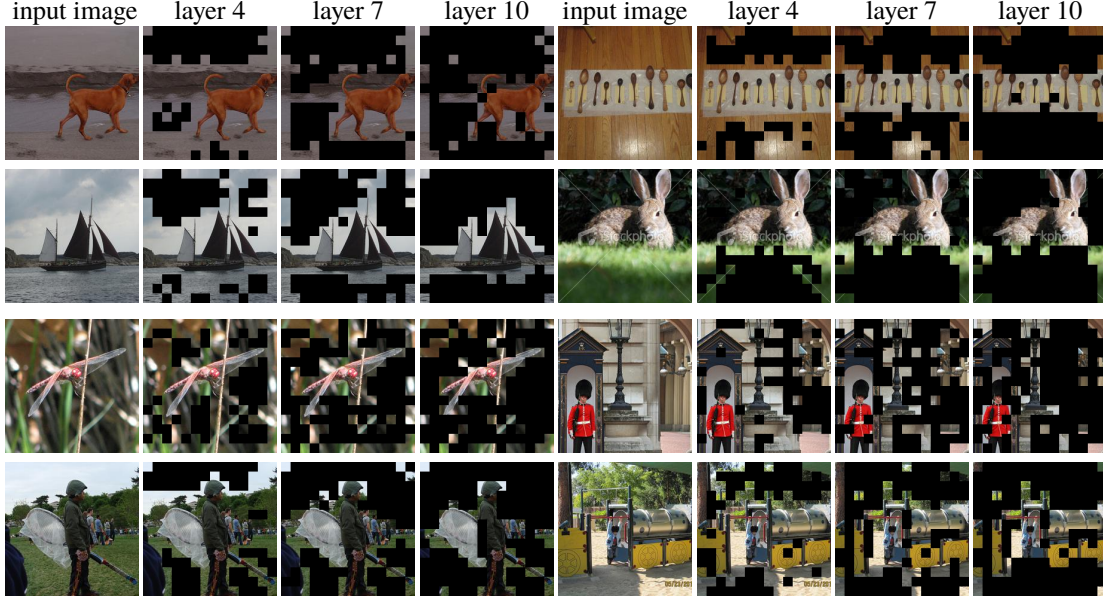


Figure 2.3: Visualization of inattentive tokens on EViT-DeiT-S with 12 layers. The masked regions represent the inattentive tokens that are fused into a new token. Our method can effectively identify inattentive tokens in images with either simple inattentive tokens (in the first two rows) or complex inattentive tokens (in the last two rows).

2.5 Experiments

Implementation details. We train all of the models on the ImageNet (Deng et al., 2009) training set with approximately 1.2 million images and report the accuracy on the 50k images in the test set. By default, the token identification module is incorporated into the 4th, 7th and 10th layer of DeiT-S and DeiT-B (with 12 layers in total) and incorporated into the 5th, 9th and 13th layer of LV-ViT-S (with 16 layers in total). The image resolution in training and testing is 224×224 unless otherwise specified. For the training strategies and optimization methods, we simply follow those in the original papers of DeiT (Touvron et al., 2021a) and LV-ViT (Jiang et al., 2021). Since our method can be easily incorporated into these models without making substantial modifications to them, the original training strategies work well with our method. Besides, we adopt a warmup strategy for attentive token identification. Specifically, the keep rate of attentive tokens is gradually reduced from 1 to the target value with a cosine schedule.

Table 2.2: Comparison on the two variants of EViT on DeiT-S (Touvron et al., 2021a). The values of the Top-1 and Top-5 accuracy are averaged over three independent trials. The number behind $\pm$ is the standard deviation of the three trials. Each parenthetical number is the gap of the corresponding value w.r.t. the baseline DeiT-S.

Keep rate	Top-1 Acc (%)	Top-5 Acc (%)	Throughput (images/s)	MACs	Top-1 Acc (%)	Top-5 Acc (%)	Throughput (images/s)	MACs
DeiT-S	79.8	94.9	2923	4.6	79.8	94.9	2923	4.6
EViT without inattentive token fusion					EViT with inattentive token fusion			
0.9	79.9 $\pm$ 0.1 (+0.1)	94.9 $\pm$ 0.0 (-0.0)	3201 (+10%)	4.0 (-13%)	79.8 $\pm$ 0.1 (-0.0)	95.0 $\pm$ 0.0 (+0.1)	3197 (+9%)	4.0 (-13%)
0.8	79.7 $\pm$ 0.0 (-0.1)	94.8 $\pm$ 0.0 (-0.1)	3772 (+27%)	3.5 (-24%)	79.8 $\pm$ 0.0 (-0.0)	94.9 $\pm$ 0.0 (-0.0)	3619 (+24%)	3.5 (-24%)
0.7	79.4 $\pm$ 0.1 (-0.4)	94.7 $\pm$ 0.1 (-0.2)	4249 (+45%)	3.0 (-35%)	79.5 $\pm$ 0.0 (-0.3)	94.8 $\pm$ 0.0 (-0.1)	4385 (+50%)	3.0 (-35%)
0.6	79.1 $\pm$ 0.2 (-0.7)	94.5 $\pm$ 0.1 (-0.4)	4967 (+70%)	2.6 (-43%)	78.9 $\pm$ 0.1 (-0.9)	94.5 $\pm$ 0.0 (-0.4)	4722 (+62%)	2.6 (-43%)
0.5	78.4 $\pm$ 0.2 (-1.4)	94.1 $\pm$ 0.0 (-0.8)	5325 (+82%)	2.3 (-50%)	78.5 $\pm$ 0.0 (-1.3)	94.2 $\pm$ 0.0 (-0.7)	5408 (+85%)	2.3 (-50%)

Table 2.3: Results of EViT on LV-ViT-S

Keep rate	Top-1 Acc (%)	Top-5 Acc (%)	Throughput (images/s)	MACs (G)
LV-ViT-S	83.3	–	2112	6.6
0.7	83.0 (-0.3)	96.3	2954 (+40%)	4.7 (-29%)
0.5	82.5 (-0.8)	96.2	3603 (+71%)	3.9 (-41%)

Table 2.4: Results of training EViT-DeiT-S with a keep rate of 0.7 for different epochs.

Keep rate	Epochs	Top-1 Acc (%)	Top-5 Acc (%)	MACs (G)
0.7	300	79.5	94.8	3.0
0.7	450	80.2	95.1	3.0
0.7	600	81.0	95.3	3.0

Unlike DynamicViT (Rao et al., 2021), we do not use a pretrained ViT to initialize our models in most experiments, except in the experiments with an oracle ViT (see the following paragraphs). We train the models with EViT from scratch for 300 epochs on 16 NVIDIA A100 GPUs and measure the throughput of the models on a single A100 GPU with a batch size of 128 unless otherwise specified. The multiply-accumulate computations (MACs) metric is measured by `torchprofile` (Liu, 2021).

We report the main results of EViT on Tables 2.2 and 2.3. On both DeiT and LV-ViT, the proposed EViT achieves significant speedup while restricting the accuracy drop in a relatively small range. For example, DeiT-S trained with EViT with a keep rate of 0.7 increase the inference throughput by 50% while maintaining the Top-1 accuracy reduction within 0.3% on ImageNet.

Inattentive token fusion. To mitigate the problem of information loss in token removal, we propose inattentive token fusion, which fuses the non-topk tokens according to the attentiveness from the [CLS] token. We experimentally compare the proposed token reorganization method with and without inattentive token fusion. As shown in Table 2.2, token reorganization with

inattentive token fusion typically outperforms the one without inattentive token fusion. Although the improvement is small, there is no additional computational overhead introduced. From another perspective, it indicates the effectiveness of our attentive token identification because the majority of effective tokens are well preserved. Moreover, the accuracy fluctuation in EViT with inattentive token fusion is smaller than the vanilla inattentive token removal, as shown by the standard deviation in Table 2.2. Inattentive token fusion also benefits EViT on high resolution images, as shown in Table 2.9.

Epochs of training. Since ViTs do not have inductive bias such as translational invariance processed by CNNs, they typically require more training data and/or training epochs to reach a comparable generalization performance as CNNs (Dosovitskiy et al., 2021; Touvron et al., 2021a). We find that training longer epochs continues to benefit ViTs in the efficient computation regime. We train the DeiT-S model with 0.7 keep rate for longer epochs of 450 and 600, respectively. As shown in Table 2.4, their performance steadily improves with training epochs.

Training/Finetuning on higher resolution images. By fusing the inattentive tokens, We are able to feed EViT with more tokens under the same computational cost. Therefore, we train and/or finetune EViT on resized images with higher resolutions than the standard resolution of 224^2 , and we report the results on Table 2.5. We can see that EViT in most tested cases performs favourably against a vanilla DeiT/LV-ViT, while having comparable or higher inference throughput than the baselines. Notably, training EViT-LV-ViT-S on images of resolution 224^2 and further finetuning it on a higher resolution of 448^2 for another 100 epochs gives a very competitive Vision Transformer model, achieving an ImageNet top-1 accuracy of 84.7%, which is 0.3% higher than LV-ViT-S@384 with basically the same throughput and number of parameters. The experimental results validate our hypothesis that images typically contain tokens that are less informative and contribute little to the recognition task. Since ViTs perform global self-attention among all tokens in as early as the first layer, the early interaction and information exchange between tokens makes it possible to discard/fuse some of the least informative tokens in intermediate ViT layers since they have been “seen” by other tokens (including the [CLS] token). In comparison, the receptive

field of a CNN at the shallow layers is relatively small due to the locality property of convolution, which makes it difficult to reduce computation at early stages.

Training with an oracle ViT. In EViT, the criterion of selecting tokens is the attention between the [CLS] token and other tokens. Therefore, it would be very helpful if we know the importance of each token to the prediction tasks in advance. To this end, we introduce an oracle ViT to guide the [CLS] through the token selection process. A good oracle knows which tokens are important and which are not. For this purpose, we use a fully trained DeiT-S/B as an oracle and initialize EViT-DeiT-S/B with the parameters of the oracle ViT, such that the EViT know which tokens contribute more to the prediction result. We train EViT equipped with an oracle using the same training setup as training a vanilla EViT. As shown in Table 2.6, training with an oracle significantly improves the recognition accuracy of EViT.

Comparison with DynamicViT. We compare the proposed EViT with DynamicViT (Rao et al., 2021). Note that the authors of DynamicViT used a finetuning strategy. Therefore, we first compare the two methods in finetuning, and then we also train DynamicViT from scratch for 300 epochs to compare it with our baselines. In the case of training from scratch, no pretrained ViT can be used, and thus the distillation loss and KL loss in DynamicViT are discarded. The results in Table 2.7 show that EViT outperforms DynamicViT in recognition accuracy under the same computational cost while EViT uses fewer parameters. Moreover, the accuracy of EViT continues to improve in longer-epoch training. When training from scratch, a significant accuracy reduction is observed in DynamicViT, which suggests that DynamicViT requires a teacher ViT (i.e., the parameter initialization, the distillation loss, and the prediction (KL) loss from a pretrained ViT (Rao et al., 2021)) to obtain a good performance.

Comparison with other Vision Transformers. We plot the accuracy and throughput of various ViTs in Figure 2.4, which shows EViT is very competitive among the vision models in terms of computation-accuracy trade-off. For EViT, we only plot the EViT models that are trained from scratch for fair comparisons, except EViT/0.7-LV-ViT-S@448, which is finetuned for 100 epochs. The comparing methods include DeiT (Touvron et al., 2021a), CaiT (Touvron et al.,

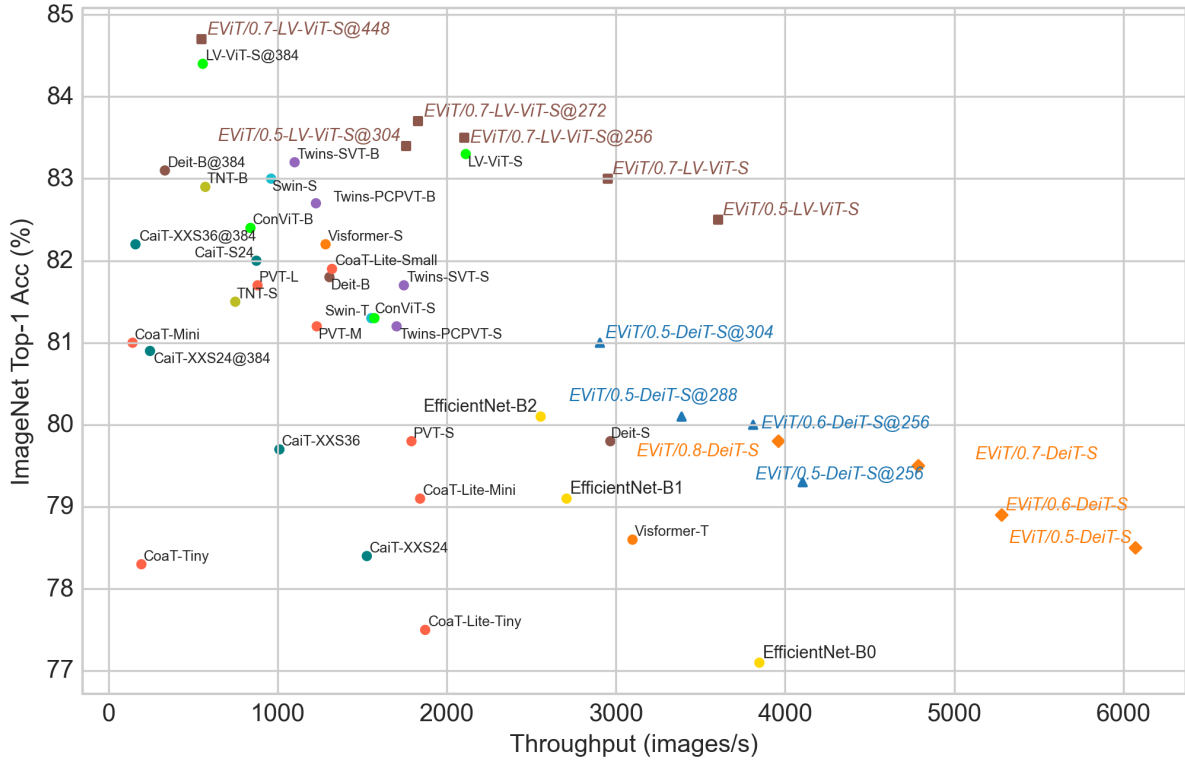


Figure 2.4: Comparison of different models with various accuracy-throughput trade-off. The proposed method EViT achieves better trade-off than the other methods (marked with circles). The throughput is measured on an NVIDIA A100 GPU using the largest possible batch size for each model. The input image size is 224^2 unless specified after the @.

2021b), LV-ViT (Jiang et al., 2021), CoaT (Xu et al., 2021), Swin (Liu et al., 2021), Twins (Chu et al., 2021), Visformer (Chen et al., 2021), ConViT (d’Ascoli et al., 2021), TNT (Han et al., 2021a), EfficientNet (Tan and Le, 2019), and PVT (Wang et al., 2021a). These ViTs variants focus on modifying the ViT architectures or the interaction methods between image tokens to achieve improvement over the vanilla ViT (Dosovitskiy et al., 2021). Thus, their methods and ours are complementary to each other, which makes it possible to incorporate our method into some of the ViT variants.

2.6 Discussion

In this chapter, we present a token reorganization method. By identifying the tokens with the largest attention from the class token, the proposed EViT reaches a better trade-off between accuracy and efficiency than various Vision Transformer models. Moreover, we propose inattentive token fusion, which fuses the information from less informative tokens to a new token. Inattentive token fusion improves both the recognition accuracy and training stability. Experimentally, we apply the proposed token reorganization method to two variants of Vision Transformer, namely, DeiT (Touvron et al., 2021a) and LV-ViT (Jiang et al., 2021). In both variants, EViT achieves a significant speedup in inference while the reduction in recognition accuracy is relatively small. Moreover, when training on higher resolution images, EViT improves the accuracy to various extents while maintaining a similar or smaller computation cost as the original DeiT/LV-ViT models. Besides, when equipped with an oracle ViT which knows which tokens are more important, EViT can achieve further improvement on the trade-off between accuracy and efficiency. The proposed EViT can be easily adapted in ViTs and brings no additional parameters, nor does it require sophisticated training strategies. The proposed token reorganization method can serve as an effective acceleration approach for Vision Transformers.

2.7 Additional Visualization

In this part, we present more visualization results in Figure 2.5 to show the attentive token identification. The input images are randomly selected from the ImageNet dataset. The results validate that our EViT is able to deal with different images from various categories.

2.8 Extended Experiments and Ablations

More results of EViT on DeiT. We provide the performance of EViT on DeiT-B (Touvron et al., 2021a) in Table 2.8. Again, the resulting models of EViT-DeiT-B achieve significant speedup while restricting the accuracy drop in a relatively small range. For example, EViT-DeiT-B with a

Algorithm 1: PyTorch-like pseudocode of EViT for a ViT encoder.

```
# H: number of attention heads
# N: number of input tokens
# C: the dimension of token vector
# k: the token keep rate
# x: the input tokens with shape [N, C], with the first being the [CLS] token
# fc_q, fc_k, fc_v: linear transforms for query, key, and value of self-attention
# @: matrix multiplication
# proj: linear projection in self-attention
# norm: layer normalization
# ffn: feed-forward network

# initialize
avg_cls_attn = zeros(N-1)
x_out = []
x_residual = x
x = norm(x)

## multi-head self-attention computation
for i in range(0, H): # compute self-attention for each attention head
    # linearly map the tokens to query, key, and value matrices
    q, k, v = fc_q[i](x), fc_k[i](x), fc_v[i](x)
    # compute the attention map
    attn = (q @ k.transpose()) / sqrt(C/H)
    attn = softmax(attn, dim=1)

    x_head = attn @ v
    x_out.append(x_head)

    # compute the [CLS] attentiveness w.r.t. other tokens,
    # without using [CLS] attention to itself
    cls_attn = attn[0, 1:]
    avg_cls_attn += cls_attn

# concatenate the output tokens of all heads
x = concat(x_out, dim=1)

x = proj(x) # shape: [N, C]
x = x + x_residual

# average the [CLS] attentiveness over all heads
avg_cls_attn /= H

# sort the avg_cls_attn in descending order
sorted_cls_attn, idx = sort(avg_cls_attn)

# compute the number of attentive tokens, without counting the [CLS] token
K = ceil(k * (N - 1))

topk_attn, topk_idx = sorted_cls_attn[:K], idx[:K]
non_topk_attn, non_topk_idx = sorted_cls_attn[K:], idx[K:]

# separate [CLS] token and other tokens
cls_token = x[0:1]
x_without_cls = x[1:]

# obtain the attentive and inattentive tokens
attentive_tokens = x_without_cls[topk_idx]
inattentive_tokens = x_without_cls[non_topk_idx]

# compute the weighted combination of inattentive tokens
fused_token = non_topk_attn @ inattentive_tokens

# concatenate these tokens
x_new = concat([cls_token, attentive_tokens, fused_token], dim=0)

x_residual = x_new
x_new = norm(x_new)
x_new = ffn(x_new)
x_new = x_new + x_residual
return x_new
```

keep rate of 0.8 increases inference throughput by 26% while maintaining the Top-1 accuracy reduction within 0.1% on ImageNet.

More comparisons of vanilla inattentive token removal and inattentive token fusion.

We compare the performance of the two variants of EViT, namely, vanilla inattentive token removal and inattentive token fusion, on the training of higher resolution images. In this scenario, EViT with inattentive token fusion also outperforms the EViT with vanilla inattentive token removal, especially in the Top-5 accuracy, as shown in Table 2.9. Although the improvement is relatively small, it is practical as inattentive token fusion brings no additional computational overhead compared to vanilla inattentive token removal.

Using tokens-to-tokens attentive scores for inattentive tokens identification. As discussed in Section 2.4.2, we use the attention from the [CLS] token to other tokens as the attentive scores to identify the class-related tokens that are important to visual recognition. This attentive/inattentive token identification approach is motivated by the fact that the final prediction of ViTs is determined by the representation of the [CLS] token in the last layer. However, we also find that it is possible to use a tokens-to-tokens attention approach to identify the attentive tokens, where the attention scores are computed as the average of the attention from all tokens to all tokens. Specifically, recall the attention map A is computed by:

$$A = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d}}\right). \quad (2.3)$$

The attentive score is computed by

$$a = \frac{1}{n} \sum_{i=1}^n A_{i,:} \quad (2.4)$$

where n is the number of tokens and $A_{i,:}$ is the i -th row of the attention map A . When there are multiple self-attention heads, the final attentive score is averaged over all heads: $\bar{a} = \sum_{h=1}^H a^{(h)} / H$, where H is the number of heads. In a nutshell, in the tokens-to-tokens strategy, each token can “vote” to decide which tokens are attentive, while in the [CLS]-to-tokens strategy, only the

[CLS] token can “vote”. As shown in Table 2.10, EViT with the tokens-to-tokens attention strategy performs comparably with the [CLS]-to-tokens attention strategy but is less efficient. Nevertheless, this experimental observation opens the door for incorporating the proposed EViT into the ViT variants with no [CLS] tokens (e.g., PVT (Wang et al., 2021a)).

Training time of EViT. We provide a comparison on the training time of DeiT-S (Touvron et al., 2021a) and EViT-DeiT-S in Table 2.11. The results show that EViT requires considerably smaller GPU×hours to train when compared to a vanilla DeiT under the same number of epochs. Since EViT continues to improve in recognition accuracy with larger training epoch (as shown in Table 2.4), the performance gap between a vanilla ViT and an EViT can be narrowed down and even closed when the EViT is trained for the same amount of time as the ViT.

Finetuning of EViT on high resolution images. We provide results of finetuning EViT on high-resolution images in the last three rows in Table 2.12. To finetune EViT-DeiT-S from low-resolution images to high-resolution images, we need to resolve the discrepancy between the positional embedding (Touvron et al., 2021a) of two different resolutions. To this end, we interpolate the positional embedding of lower resolution to the desired higher resolution. This method works quite well in our experiments, as shown in the last three rows in Table 2.12. However, directly training on higher resolution images seems a better way to train EViT, as it requires less training time (because no finetuning is performed) and typically obtains higher accuracy than finetuning.

Token reorganization locations. It is possible that different combinations of keep rates and token reorganization locations can reach the same computational efficiency. For example, we can a) move the reorganization modules into shallower layers and increase the keep rates, or b) move the reorganization modules into deeper layers and decrease the keep rates, to keep the computation cost (approximately) unchanged. To get a deeper understanding of this aspect, we train EViT-DeiT-S models with different token reorganization locations and keep rates, each of which has a similar computational cost as the baseline. The results in Table 2.13 reveal two perspectives. First, moving the reorganization modules into shallower layers deteriorates the

accuracy. For example, when the token reorganization module is placed before the third layer (i.e., at the first or second layer), the recognition accuracy drops considerably even though the computational cost is the same. This suggests that ViT cannot identify the important tokens at the early stages, which is quite reasonable as the processing of input tokens is just started at shallow layers, where the attention maps are unreliable for token removal. Second, placing the reorganization modules in different deeper layers has only marginal influence on the accuracy. For instance, when the token reorganization modules are placed behind the third layer, the resulting models have basically the same accuracy. This suggests that EViT has stable performance as long as the reorganization locations are not in shallow layers. Because of this reason, we refrain from searching for a (slightly) better configuration of the reorganization locations and keep rates. We adopt a simple strategy to decide the reorganization locations in our experiments, where the token reorganization locations cut the ViT into blocks with the same number of layers. Specifically, for a ViT with L layers and t token reorganization layers in total, we first obtain the separating length $s = L/(t + 1)$. Then, the layer indices of the reorganization layers are $[s + 1, 2s + 1, \dots, ts + 1]$, which cuts the ViT evenly. For the keep rates, we simply set them to the same value for each token reorganization module in the EViT.

Adopting the DINO attention map for inattentive token identification. In DINO (Caron et al., 2021), the authors showed that the [CLS] attention produced by a self-supervised trained ViT is naturally focused on the object parts in images. Therefore, one may wonder if the good DINO attention can be used in EViT for inattentive token identification. To answer this question, we use the [CLS] attention produced by the last layer of DINO to guide the token selection process in EViT. Instead of starting the token reorganization process at the 4th layer of EViT-DeiT-S as we have done in previous experiments, we start the token reorganization at the first layer. Specifically, we select the top 50% tokens that correspond to the positions with the strongest DINO attention (i.e., corresponding to the top 50% attention values in the DINO attention). Experimental results in Table 2.14 show that EViT works well with the DINO attention. As a comparison, when the reorganization layer is placed in the first layer in a vanilla EViT, as

shown in Table 2.13, a significant accuracy reduction is observed, while EViT equipped with DINO can maintain its recognition ability even if 50% of tokens are removed as early as the first layer. However, this is not an efficient approach in practice because obtaining the DINO attention requires forwarding the input through the DINO model (i.e., a DeiT model) and bringing extra computational burden, as shown by the “Combined MACs” in Table 2.14.

The correlation between the quality of the attention masks and the performance of EViT. From Figures 2.3 and 2.5, we can see that the masks produced by EViT are quite intuitive and of good quality, which means that the attention values of the CLS token to other tokens correlate well with the semantic segmentation of images. Therefore, we can deliberately use low quality masks in EViT to see how its performance changes with the quality of the masks. Note that we use the largest [CLS] -to-tokens attention values to obtain the mask for the keeping tokens (we call this method EViT-topk). Thus, the worse possible mask one can obtain is probably using the smallest [CLS] -to-tokens attention values (we call this method EViT-mink), and a not-so-bad mask can be obtained by randomly generating a mask without considering the [CLS] -to-tokens attention (we call this method EViT-random). We perform experiments to see the classification accuracy of EViT under these circumstances. As shown in Table 2.15, the classification accuracy of these three methods clearly correlates with the quality of the masks. From the experimental results, we conclude that masks of better quality help improve the accuracy. Therefore, we believe it is a potential direction for future research as to how to (efficiently) obtain a better mask to improve the performance of EViT.

The attention scores of the inattentive tokens w.r.t. the layer depth. In order to find how the attentive scores of the inattentive tokens evolve, we randomly sample several images and plot the change of the [CLS] -to-tokens attention values of n inattentive tokens ($n = 25$) w.r.t. the depth of layers in Fig. 2.6. Specifically, we use a pre-trained DeiT-S and identify the n tokens with the smallest [CLS] -to-tokens attention in the 4th layer. Then we keep track of these selected tokens in the subsequent layers and draw the box plot to see how the statistics (e.g., the mean, standard deviation, and maximum and minimum values) change with the layer depth. The

results suggest that the tokens that are identified as inattentive tokens in the shallow layer tend to be less preferred in deeper layers.

The evolution of the inattentive tokens during EViT training process. To prove the effectiveness of incorporating EViT into the training process of ViT, we present the evolution of the inattentive tokens during the training of EViT-DeiT-S with a keep rate of 0.7 in Figure 2.7. It can be observed that at the initial training stage, the EViT model tends to misidentify the inattentive tokens (e.g., the bulk of the car in the second column of the first three rows). However, the inattentive tokens gradually converge to the less informative regions of images during the training process. For example, in the last column in Figure 2.7, the masked regions mainly focus on the background (e.g., the grass). This observation validates the effectiveness of our proposed token identification methods.

2.9 From Adaptive Computation to Adaptive Prediction

EViT demonstrates that deep models can become more efficient by adaptively selecting and reorganizing input information. However, in many scientific applications, the challenge is not only deciding which input components to process, but also deciding how to represent heterogeneous objects and how to support diverse prediction tasks. Molecular property prediction provides such a setting: a molecule can be represented as a graph, an image, a SMILES string, or text, and the desired output may range from a binary activity label to a free-form description of mechanism of action. The next chapter extends the principle of adaptation from input processing to multimodal representation and prompt-conditioned prediction.

Chapter 2, in full, is a reprint of the material as it appears in “Not All Patches are What You Need: Expediting Vision Transformers via Token Reorganizations” by Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie, which was published in International Conference on Learning Representations as a Spotlight paper in 2022. The dissertation author was the primary researcher and author of this paper.

Table 2.5: Results of training/finetuning on high resolution images. EViT-DeiT-S and EViT-LV-ViT-S have a throughput/MACs comparable to the baselines while achieving better recognition accuracy on ImageNet. The numbers before and after $\uparrow$ indicate the image size in the 300-epoch training and 100-epoch finetuning, respectively.

(a) DeiT-S							(b) LV-ViT-S						
Model	Keep rate	Image size	Top-1 (%)	Top-5 (%)	img/s	MACs (G)	Model	Keep rate	Image size	Top-1 (%)	Top-5 (%)	img/s	MACs (G)
DeiT-S	1.0	224	79.8	94.9	2923	4.6	LV-ViT-S	1.0	224	83.3	–	2112	6.6
EViT	0.5	256	79.3	94.7	3788	3.1	LV-ViT-S	1.0	224 $\uparrow$ 384	84.4	–	557	21.9
EViT	0.5	288	80.1	95.0	3138	3.9	EViT	0.9	240	83.6	96.5	1956	6.8
EViT	0.5	304	81.0	95.6	2905	4.4	EViT	0.8	256	83.6	96.6	1901	6.9
EViT	0.6	256	80.0	95.0	3524	3.5	EViT	0.7	256	83.5	96.5	2102	6.2
EViT	0.6	288	81.0	95.4	2927	4.5	EViT	0.7	272	83.7	96.6	1829	7.1
EViT	0.7	272	80.3	95.3	2870	4.6	EViT	0.5	304	83.4	96.5	1758	7.4
							EViT	0.7	272 $\uparrow$ 448	84.7	97.1	548	21.5

Table 2.6: Results of training EViT-DeiT-S and EViT-DeiT-B using DeiT-S as an oracle. Training for longer epochs continues to benefits the EViT in efficiency regime.

Model	Oracle	Keep rate	Epochs	Top-1 (%)	Top-5 (%)	Throughput (img/s)	MACs (G)
DeiT-S	$\times$	1.0	300	79.8	94.9	2923	4.6
EViT-DeiT-S	$\times$	0.7	300	79.5	94.8	4385	3.0
EViT-DeiT-S	$\checkmark$	0.7	300	80.8	95.4	4385	3.0
EViT-DeiT-S	$\checkmark$	0.7	450	81.0	95.5	4385	3.0
EViT-DeiT-S	$\checkmark$	0.7	600	81.3	95.5	4385	3.0
DeiT-B	$\times$	1.0	300	81.8	95.6	1295	17.6
EViT-DeiT-B	$\times$	0.7	300	81.3	95.3	2053	11.6
EViT-DeiT-B	$\checkmark$	0.7	300	82.1	95.6	2053	11.6

Table 2.7: Comparisons on EViT and DynamicViT (Rao et al., 2021). ✓ indicates the model is initialized with a pre-trained DeiT-S. For fair comparison, the throughput is measured on the same machine with the same setting using a maximum batch size.

Model	Pre-trained	Epochs	Keep rate	Top-1 (%)	#Params (M)	Throughput (img/s)	MACs (G)
DynamicViT-DeiT-S	✓	30	0.5	77.5	22.8	5579	2.2
EViT-DeiT-S	✓	30	0.5	78.5 (+1.0)	22.1 (-0.7)	5549	2.3
EViT-DeiT-S	✓	100	0.5	79.1 (+1.6)	22.1 (-0.7)	5549	2.3
DynamicViT-DeiT-S	✓	30	0.7	79.3	22.8	4439	3.0
EViT-DeiT-S	✓	30	0.7	79.5 (+0.2)	22.1 (-0.7)	4478	3.0
EViT-DeiT-S	✓	100	0.7	79.8 (+0.5)	22.1 (-0.7)	4478	3.0
DynamicViT-DeiT-S	×	300	0.5	73.1	22.8	5579	2.2
EViT-DeiT-S	×	300	0.5	78.5 (+5.4)	22.1 (-0.7)	5549	2.3
DynamicViT-DeiT-S	×	300	0.7	77.6	22.8	4439	3.0
EViT-DeiT-S	×	300	0.7	79.5 (+1.9)	22.1 (-0.7)	4478	3.0

Table 2.8: Results of EViT on DeiT-B

Keep rate	Top-1 Acc (%)	Top-5 Acc (%)	Throughput (images/s)	MACs (G)
DeiT-B	81.8	95.6	1295	17.6
0.9	81.8 (-0.0)	95.6 (-0.0)	1441 (+11%)	15.3 (-13%)
0.8	81.7 (-0.1)	95.4 (-0.2)	1637 (+26%)	13.2 (-25%)
0.7	81.3 (-0.5)	95.3 (-0.3)	2053 (+59%)	11.5 (-35%)
0.6	80.9 (-0.9)	95.1 (-0.5)	2177 (+68%)	10.0 (-43%)
0.5	80.0 (-1.8)	94.5 (-1.1)	2482 (+92%)	8.7 (-51%)

Table 2.9: Comparisons of the two variants (i.e., vanilla inattentive token removal and inattentive token fusion) of EViT-DeiT-S on higher resolution images.

Method	Keep rate	Image size	Top-1 Acc (%)	Top-5 Acc (%)	Throughput (images/s)	MACs (G)
DeiT-S	1.0	224	79.8	94.9	2923	4.6
Inattentive token removal	0.5	304	80.9 (+1.1)	95.3 (+0.4)	2927	4.4
Inattentive token fusion	0.5	304	81.0 (+1.2)	95.6 (+0.7)	2905	4.4

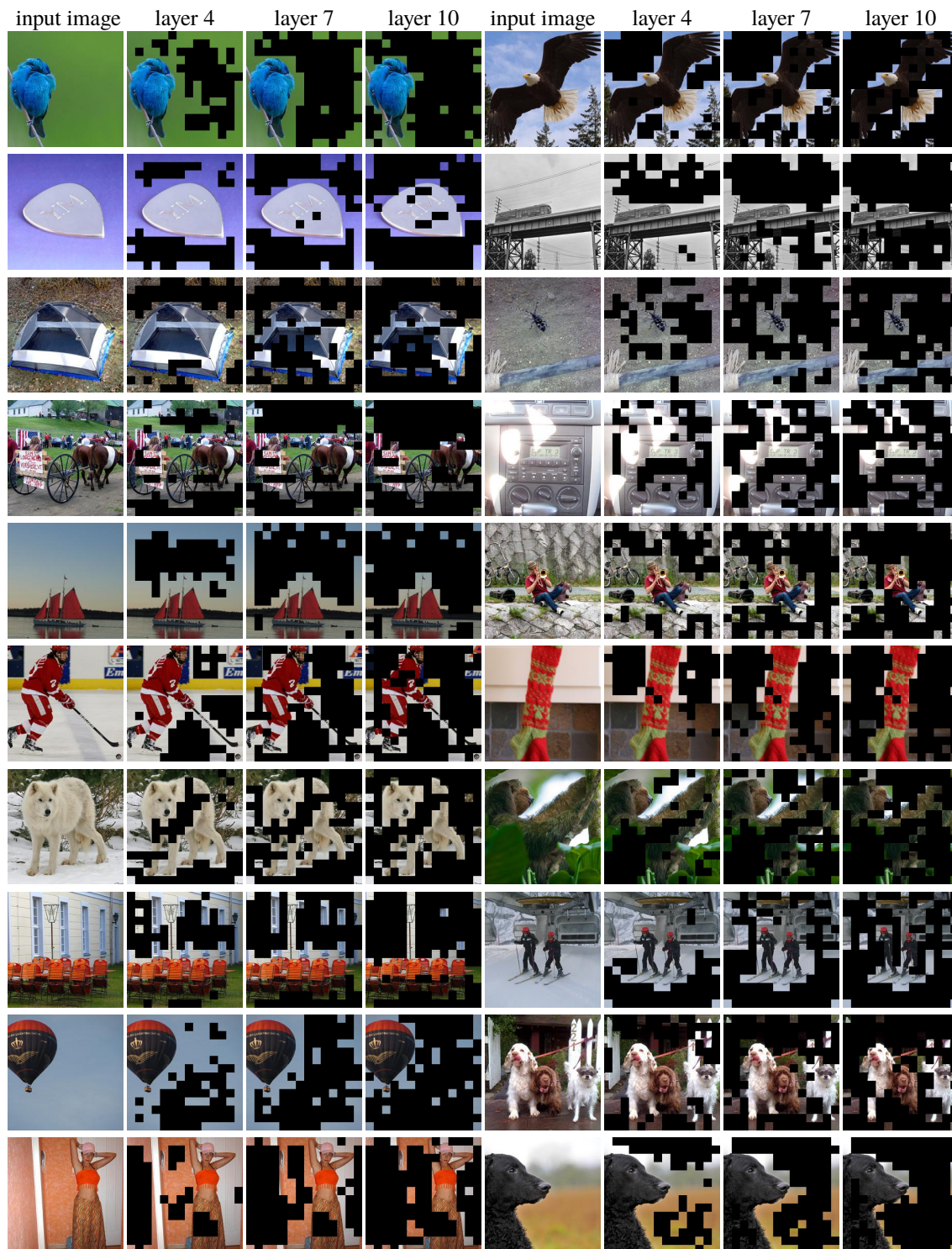


Figure 2.5: Extended visualizations of token reorganization in 12-layer EViT-DeiT-S. Unmasked regions are attentive tokens; masked regions are inattentive tokens fused into a new token. The examples span diverse ImageNet categories.

Table 2.10: The performance of EViT-DeiT-S with different attentive token identification strategies.

Method	Inattentive token fusion	Keep rate	Top-1 Acc (%)	Top-5 Acc (%)	Throughput (img/s)	MACs (G)
[CLS]-to-tokens attentiveness	×	0.5	78.4	94.1	5325	2.3
	✓	0.5	78.5	94.2	5408	2.3
	×	0.7	79.4	94.7	4249	3.0
	✓	0.7	79.5	94.8	4385	3.0
Tokens-to-tokens attentiveness	×	0.5	78.4	94.2	5241	2.3
	✓	0.5	78.5	94.3	5346	2.3
	×	0.7	79.4	94.8	4152	3.0
	✓	0.7	79.4	94.7	4290	3.0

Table 2.11: Training time for 300-epoch training of DeiT-S and EViT-DeiT-S with different keep rates.

Keep rate	Top-1 Acc (%)	Top-5 Acc (%)	Throughput (images/s)	MACs (G)	Training time (GPU × hours)
DeiT-S	79.8	94.9	2923	4.6	201
0.9	79.8 (-0.0)	95.0 (+0.1)	3197 (+9%)	4.0 (-13%)	188 (-6%)
0.8	79.8 (-0.0)	94.9 (-0.0)	3619 (+24%)	3.5 (-24%)	174 (-13%)
0.7	79.5 (-0.3)	94.8 (-0.1)	4385 (+50%)	3.0 (-35%)	160 (-20%)
0.6	78.9 (-0.9)	94.5 (-0.4)	4722 (+62%)	2.6 (-43%)	150 (-25%)
0.5	78.5 (-1.3)	94.2 (-0.7)	5408 (+85%)	2.3 (-50%)	145 (-28%)

Table 2.12: Results of training/finetuning on high resolution images. The numbers before and after ↑ indicate the image size in the 300-epoch training and 50-epoch finetuning, respectively.

Model	Keep rate	Image size	Top-1 (%)	Top-5 (%)	Throughput (img/s)	MACs (G)
DeiT-S	1.0	224	79.8	94.9	2923	4.6
EViT-DeiT-S	0.5	256	79.3	94.7	3788	3.1
EViT-DeiT-S	0.5	288	80.1	95.0	3138	3.9
EViT-DeiT-S	0.5	304	81.0	95.6	2905	4.4
EViT-DeiT-S	0.6	256	80.0	95.0	3524	3.5
EViT-DeiT-S	0.6	288	81.0	95.4	2927	4.5
EViT-DeiT-S	0.7	272	80.3	95.3	2870	4.6
EViT-DeiT-S	0.5	224 ↑ 304	80.3	95.2	2905	4.4
EViT-DeiT-S	0.6	224 ↑ 288	80.7	95.3	2927	4.5
EViT-DeiT-S	0.7	224 ↑ 272	80.6	95.2	2870	4.6

Table 2.13: The performances of EViT-DeiT-S with different of keep rates and reorganization locations (R.L.). For fair comparison, we keep the variants having the same level of computational cost (i.e., MACs) as our EViT baseline in the first row by tuning the keep rates.

#R.L.	Reorganization locations	Keep rates	Top-1 Acc (%)	Top-5 Acc (%)	MACs (G)
3	[4, 7, 10]	0.7	79.50	94.77	3.0
3	[5, 7, 10]	[0.64, 0.70, 0.70]	79.57 (+0.07)	94.80	3.0
3	[3, 7, 10]	[0.74, 0.70, 0.70]	79.47 (-0.03)	94.72	3.0
3	[4, 8, 10]	[0.70, 0.64, 0.70]	79.64 (+0.14)	94.86	3.0
3	[4, 6, 10]	[0.70, 0.75, 0.70]	79.54 (+0.04)	94.78	3.0
3	[6, 8, 10]	0.6	79.43 (-0.07)	94.69	3.0
3	[2, 6, 10]	0.765	78.72 (-0.78)	94.22	3.0
3	[2, 4, 6]	0.81	78.45 (-1.05)	94.26	3.0
3	[1, 5, 9]	[0.80, 0.79, 0.79]	76.34 (-3.16)	93.14	3.0
4	[4, 6, 8, 10]	0.76	79.38 (-0.12)	94.72	3.0
4	[2, 5, 8, 11]	[0.90, 0.77, 0.62, 0.50]	78.93 (-0.57)	94.52	3.0
7	[4, 5, 6, 7, 8, 9, 10]	0.85	79.39 (-0.11)	94.68	3.0

Table 2.14: The performance of EViT-DeiT-S equipped with the DINO attention. The “MACs” in the table is the computational cost of the EViT-DeiT-S model and “Combined MACs” is the sum of the MACs of both EViT-DeiT-S and the DINO model (i.e., a DeiT-S) which is used to obtain the DINO attention.

Method	Keep rate	Drop layer	Top-1 Acc (%)	Top-5 Acc (%)	MACs (G)	Combined MACs (G)
DINO-EViT-DeiT-S	0.5	1 st	78.7	94.2	2.2	6.8

Table 2.15: An ablation of token removal strategy in EViT-DeiT-S. EViT-topk is the method of keeping the tokens with the largest [CLS]-to-token attention values. EViT-random denotes the method of randomly keeping the tokens, while EViT-mink is the method of keeping the tokens with the smallest [CLS]-to-token attention values.

Method	Keep rate	Top-1 Acc (%)	Top-5 Acc (%)	MACs (G)
EViT-topk	0.5	78.5	94.2	2.3
EViT-random	0.5	77.3 (-1.2)	93.5 (-0.7)	2.3
EViT-mink	0.5	75.2 (-3.3)	92.4 (-1.8)	2.3
EViT-topk	0.7	79.5	94.8	3.0
EViT-random	0.7	78.4 (-1.1)	94.2 (-0.6)	3.0
EViT-mink	0.7	76.4 (-3.1)	93.2 (-1.6)	3.0

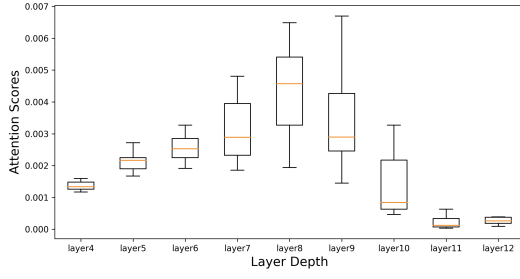


Image (a)

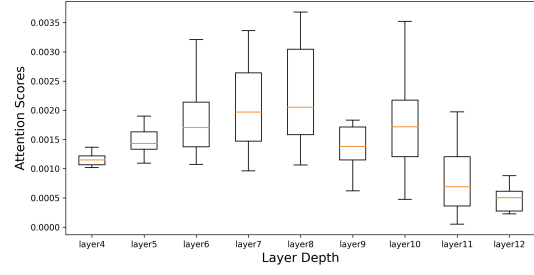


Image (b)

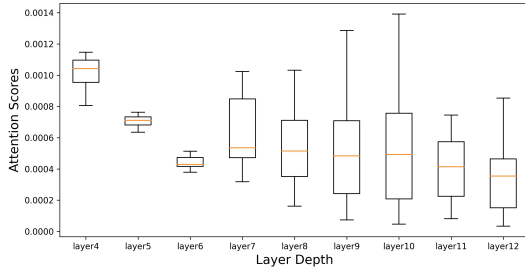


Image (c)

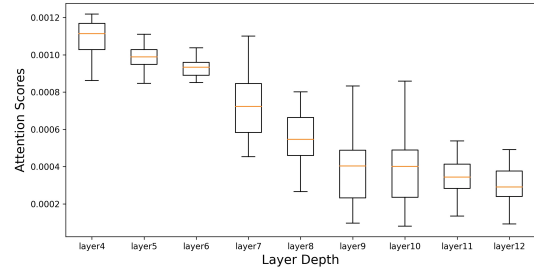


Image (d)

Figure 2.6: The box plot of the evolution of the attention scores of the inattentive tokens with the layer depth. The orange line in the box plot indicates the averaged attention scores of the inattentive tokens at a certain layer.

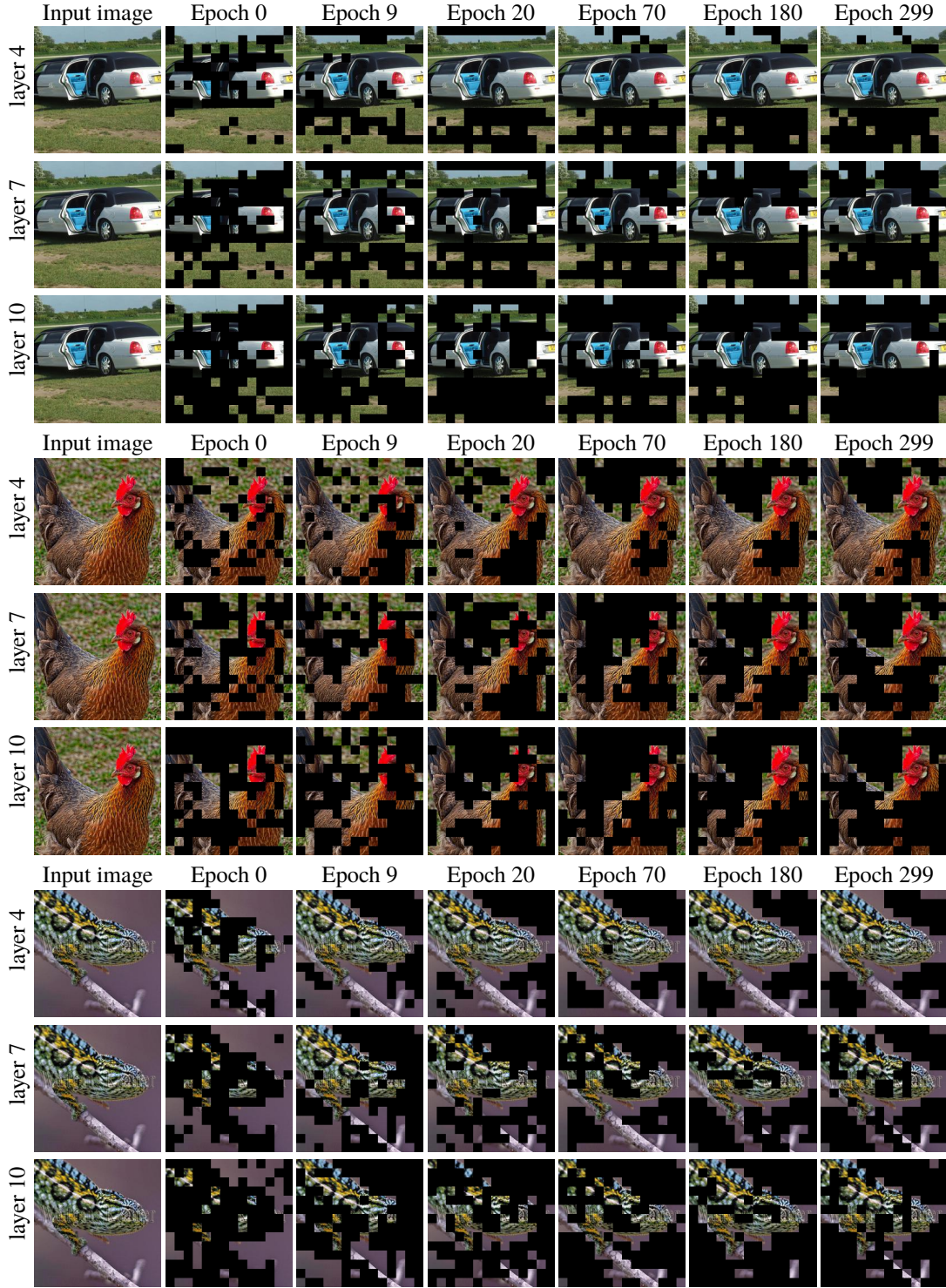


Figure 2.7: Evolution of token reorganization across EViT training epochs and layers. Unmasked regions are attentive tokens; masked regions are inattentive tokens fused into a new token. Early masks are unstable but converge to meaningful regions as training proceeds.

Chapter 3

DrugChat: Adaptive Multimodal Alignment for Molecular Properties Prediction

3.1 Chapter Overview

This chapter presents DrugChat as adaptive multimodal and prompt-conditioned prediction. Whereas Chapter 2 adapts computation over image tokens, DrugChat adapts the prediction interface itself: the same molecule can be queried through different prompts, represented through multiple modalities, and answered through either structured labels or free-form text.

3.2 Motivation

Accurate prediction of the mechanisms and properties of small molecules is crucial for advancing pharmaceutical research and facilitating drug discovery (Kirchmair et al., 2015). A significant body of work has focused on developing quantitative structure-activity relationship (QSAR) models (Tropsha, 2010), which typically use classification or regression techniques to correlate the structural properties of chemical compounds with their biological activities, such as binding affinity and toxicity. QSAR models enable virtual screening of large chemical libraries (Kitchen et al., 2004), reducing the need for costly synthesis and animal testing in pharmacology and toxicology (Luechtefeld et al., 2018; Madden et al., 2020).

In recent years, deep learning (DL) has emerged as a powerful tool for QSAR modeling and drug discovery (Chen et al., 2018b; Vamathevan et al., 2019; Wieder et al., 2020; Tropsha

et al., 2024), thanks to its capacity to analyze large-scale datasets and uncover complex patterns. DL methods have improved efficiency and accuracy across multiple stages of drug discovery (Lo et al., 2018; Gupta et al., 2021), from predicting protein-ligand binding affinity (Öztürk et al., 2018; Jiménez et al., 2018; Stepniewska-Dziubinska et al., 2018) and toxicity (Mayr et al., 2016; Banerjee et al., 2018), to enabling drug repositioning (Cheng et al., 2012; Keiser et al., 2009; Gottlieb et al., 2011; Cheng et al., 2018; Chen et al., 2020). For example, graph neural networks (GNNs) such as MPNN (Gilmer et al., 2017) and Chemprop (Yang et al., 2019b) have been widely adopted for molecular property prediction, leveraging molecular graph structures as input. Multi-task deep learning approaches (Dahl et al., 2014; Unterthiner et al., 2014; Ramsundar et al., 2015; Allenspach et al., 2024; Yang et al., 2019b; Mayr et al., 2018; Jastrzębski et al., 2016; Qian et al., 2023) have demonstrated strong efficacy in simultaneously predicting ADME (Absorption, Distribution, Metabolism, Excretion) properties (Pires et al., 2015; Daina et al., 2017; Xiong et al., 2021). More recently, multi-modal models (Zeng et al., 2022b; Seidl et al., 2023; Edwards et al., 2021; Liu et al., 2024c; Xu et al., 2023) such as KV-PLM (Zeng et al., 2022b), CLAMP (Seidl et al., 2023), and Text2Mol (Edwards et al., 2021) have been proposed to jointly model molecular structures and textual descriptions of prediction tasks, combining molecular encoders with language encoders. In parallel, molecule-focused scientific large language models (LLMs) (Thirunavukarasu et al., 2023) such as Galactica (Taylor et al., 2022) have been developed to leverage large-scale molecular and textual data for scientific question answering and property prediction.

Although substantial progress has been made in deep learning approaches for molecular property prediction, existing methods still face important limitations that restrict their effectiveness and flexibility. First, many key properties of candidate drug molecules—such as indications, pharmacodynamics, and mechanisms of action—are inherently complex and context-dependent, making them better suited for description in free-form text than in rigid categorical or numerical formats. These properties often involve nuanced biological processes, therapeutic applications, and mechanistic pathways that cannot be fully captured by simple structured labels. Despite their

importance, most current methods produce structured outputs, such as discrete classes, scalar values, and dose–response curves (Yang et al., 2019a; Danhof et al., 2008; Lu et al., 2021; Ota and Yamashita, 2022; Chen et al., 2020; Zeng et al., 2019; Kuenzi et al., 2020). Consequently, they cannot generate the rich natural-language descriptions used in biomedical literature and expert annotations. Although KV-PLM (Zeng et al., 2022b), CLAMP (Seidl et al., 2023), and Text2Mol (Edwards et al., 2021) combine molecular features with textual inputs, they are designed primarily for classification or retrieval and produce similarity or likelihood scores for molecule–text pairs rather than free-form responses. Galactica (Taylor et al., 2022) supports free-form generation, but represents both SMILES and questions through a single language model rather than using dedicated molecular encoders.

In addition to their reliance on structured outputs, many existing multi-task deep learning methods for molecular property prediction (Dahl et al., 2014; Simões et al., 2018; Allenspach et al., 2024; Qian et al., 2023) face another fundamental limitation: they restrict predictions to a fixed set of tasks defined during training. These models rely on task-specific output heads, which restrict them to making predictions only for tasks they were explicitly trained on. As a result, they lack the flexibility to generalize to new, previously unseen tasks that arise at inference time — such as predicting activities for novel assays, emerging biological targets, or new therapeutic contexts. This rigidity limits their applicability, particularly in real-world drug discovery settings where continuous emergence of new prediction tasks is the norm. While multi-modal models that integrate molecular structure and natural language, such as KV-PLM (Zeng et al., 2022b), CLAMP (Seidl et al., 2023), and Text2Mol (Edwards et al., 2021), have shown promise in handling unseen tasks where predictions are structured (e.g., classification or retrieval), they are not able to handle unseen tasks where the outputs are in free-form texts.

A further limitation of current molecular prediction models is that they generate one-shot predictions without supporting interactive, multi-turn exploration. Once a model provides an output for a given input, there is no mechanism for users to iteratively refine their questions, ask follow-up inquiries, or progressively explore different aspects of a molecule within a conversa-

tional workflow. This static approach to prediction limits the depth and flexibility of molecular analysis, hindering the discovery of nuanced or context-specific insights that could be revealed through dynamic, iterative interaction.

To address these challenges, we propose DrugChat, a multi-modal large language model (LLM) framework designed to advance molecular property prediction beyond the limitations of traditional methods. First, DrugChat is capable of generating free-form textual predictions, enabling rich and nuanced descriptions of complex drug properties such as indications, pharmacodynamics, and mechanisms of action. Rather than constraining outputs to fixed categories or numerical scores, DrugChat produces natural-language responses that align with how these properties are discussed in scientific literature and expert annotations. Second, DrugChat enables open-ended, prompt-based multi-task learning. By combining a large instruction-tuned language model with specialized molecular encoders, DrugChat can transfer to prediction tasks held out from downstream fine-tuning without adding task-specific output heads. Users specify biological or pharmacological questions through flexible natural-language prompts. Furthermore, DrugChat handles both traditional QSAR tasks, such as toxicity and protein–target activity prediction, and more complex tasks, such as predicting drug indications, pharmacodynamics, and mechanisms of action, within the same framework. By modifying the input query, DrugChat dynamically switches between tasks for the same molecule (Fig. 3.1a). Finally, DrugChat enables dynamic, interactive exploration of molecules through multi-turn dialogue. Users can ask follow-up questions, refine their inquiries, and iteratively investigate different aspects of a molecule’s properties and mechanisms within a continuous conversation. Together, these capabilities provide a flexible and expressive framework for molecular property prediction and analysis.

We conducted extensive experiments to evaluate DrugChat across free-form DrugBank prediction, cytotoxicity, administration-route and prodrug prediction, MoleculeNet (Wu et al., 2018), and FS-Mol (Stanley et al., 2021). For free-form prediction, DrugChat achieved higher human-rated correctness than GPT-4 and fine-tuned Galactica. For structured QSAR tasks, DrugChat showed strong, task-dependent performance relative to GNN, multi-modal, and language-

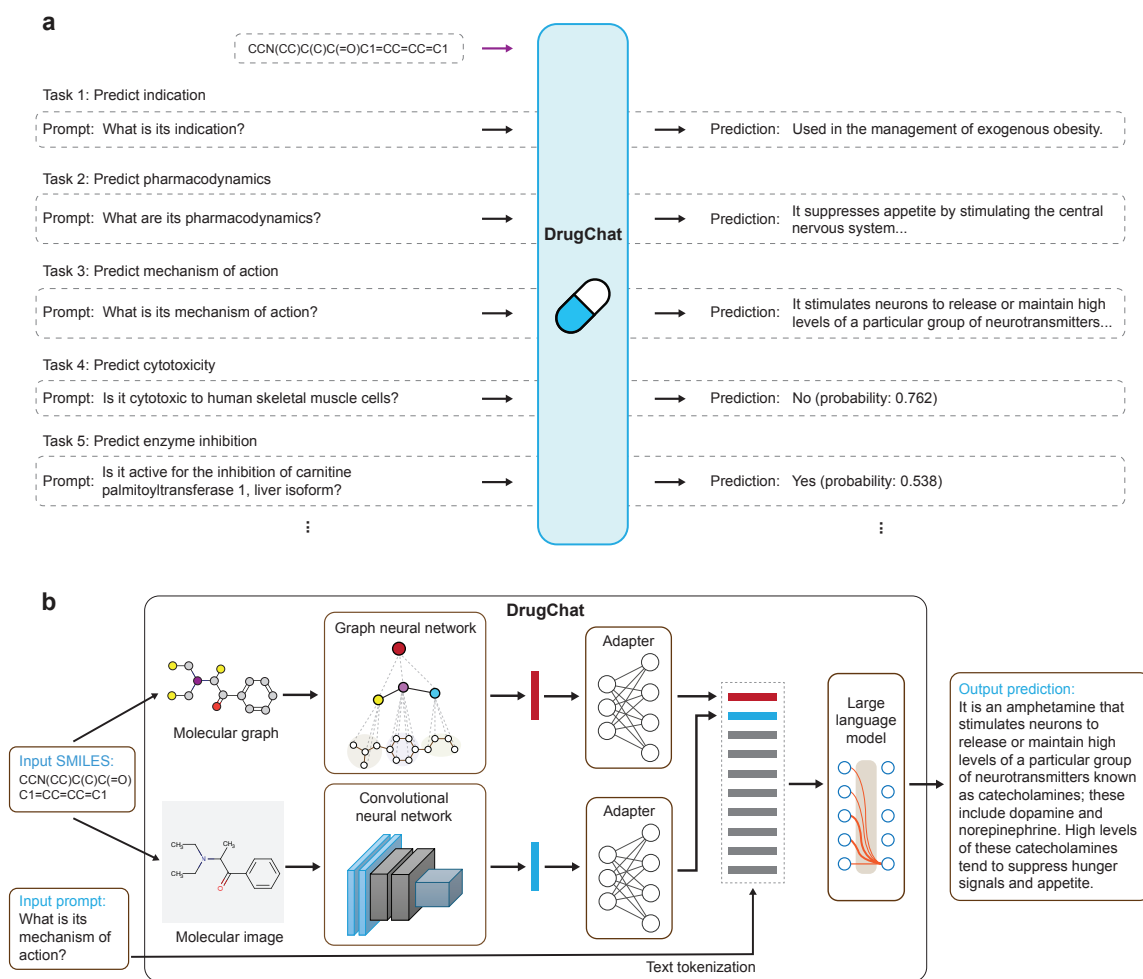


Figure 3.1: DrugChat is a multi-modal LLM capable of predicting drug properties in either free-form texts or as discrete categories. a, DrugChat facilitates versatile predictions of drug properties, enabling users to submit queries through flexible natural language (known as prompts). By using task-specific prompts, DrugChat can perform a wide range of prediction tasks within a unified framework, without requiring changes to model parameters. For classification tasks, DrugChat simultaneously outputs a probability associated with each prediction. b, Model architecture of DrugChat. It takes the SMILES representation of a molecule along with a prompt as inputs and generates a prediction in natural language. It comprises two molecular encoders — a graph neural network and a convolutional neural network — that learn representation vectors for the molecular graph and image derived from the SMILES input, two adapters that transform these representations into a format compatible with LLMs, and an LLM that generates a prediction based on the molecular representations and the prompt.

model baselines. On FS-Mol, DrugChat generalized to assays held out from downstream fine-tuning, including a stricter split that also excludes compounds appearing in the training tasks.

A human-scored study further showed that DrugChat has high context retention in interactive multi-turn dialogues. These experiments demonstrate DrugChat’s ability to handle both free-form and structured prediction tasks and to support interactive exploration of drug properties.

3.3 Method

3.3.1 Architecture and Prompt-Conditioned Prediction

DrugChat comprises a molecular graph encoder, a molecular image encoder, an instruction-tuned large language model (LLM), and two adapters that project the molecular representations into the LLM’s latent space (Fig. 3.1b). Specifically, we use a Graph Isomorphism Network (GIN) (Xu et al., 2018) as the graph encoder, a ResNet (He et al., 2016) as the image encoder, and Vicuna (Chiang et al., 2023) as the LLM. The input molecule is initially represented using a SMILES (Simplified Molecular Input Line Entry System) string. DrugChat converts the SMILES string into two forms: a structural representation as a molecular graph and a visual representation as a molecular image. The molecular graph is encoded by the GIN encoder, which is pretrained on two million unlabeled molecules from the ZINC15 database (Sterling and Irwin, 2015) using self-supervised learning (Hu et al., 2020b). The GIN processes the graph by first converting nodes and edges into distinct representation vectors based on their types. It then iteratively updates these node and edge representations by aggregating information from each node’s neighbors (Goodfellow et al., 2016; Wang et al., 2022a). Finally, an attention-based pooling operation (Wu et al., 2020) is applied to summarize the node representations into a single representation vector of the entire molecular graph. The molecular image is encoded by the ResNet encoder, which was pretrained on images of ten million unlabeled molecules from the PubChem dataset (Kim et al., 2023) using self-supervised learning (Zeng et al., 2022a). The ResNet applies multiple layers of 2D convolutional filters, which function as matching templates, to ‘scan’ the image, producing a feature map (LeCun et al., 2015). This feature map is then condensed into a representation vector of the entire molecular image through an average pooling

operation (Krizhevsky et al., 2012). The adapters, implemented as linear layers, are used to transform the representation vectors of the molecule’s graph and image into a unified vector, termed the *molecule token*, which is compatible with the LLM’s latent representation space, thereby enabling the LLM to interpret it. Concurrently, the input prompt is decomposed into a sequence of language tokens, each represented as a vector within the LLM’s latent space. The molecule token is integrated into the language token sequence, which is then fed into the LLM, specifically Vicuna-13B (Chiang et al., 2023). The LLM then generates new language tokens sequentially through autoregressive decoding (Vaswani et al., 2017b), ultimately producing the prediction. For binary classification, we derive a continuous decision score from the probabilities of the first generated “Yes” and “No” tokens (Methods). This score is used to compute Δ AUPRC but is not interpreted as a calibrated probability of correctness.

We adopt a two-stage training strategy for DrugChat: a large-scale pretraining phase followed by task-specific finetuning. The pretraining dataset, curated from PubChem (Kim et al., 2023), comprises 3.6M unique compounds and 181M activity records across 446K bioassays. During pretraining, DrugChat was trained to predict compound activity on bioassays (see Methods). This large-scale dataset enabled DrugChat to learn patterns of molecular behavior across diverse biological contexts, allowing the molecular encoders to extract informative structural features that are aligned with the language model for downstream bioactivity prediction. For finetuning, we used more than 500,000 compound-activity records across over 5,000 tasks, drawn from QSAR benchmarking datasets (including the cytotoxicity dataset (Wong et al., 2024), FS-Mol (Stanley et al., 2021), and MoleculeNet (Wu et al., 2018)) and curated subsets from ChEMBL (Mendez et al., 2019) and DrugBank (Wishart et al., 2006). These datasets cover over 230,000 unique compounds. Activity labels include both structured binary annotations (e.g., toxic vs. non-toxic) and expert-written free-form texts describing complex drug properties such as indications, pharmacodynamics, and mechanisms of action. From these annotated molecules, we curated molecule-prompt-answer triplets for training DrugChat. In these triplets, the molecule and prompt serve as inputs to DrugChat, while the answer represents the target

output of DrugChat. The answers range from free-form texts to binary responses such as ‘yes/no’ to questions like ‘can it be taken orally?’. We trained model weights by minimizing the negative log-likelihood (Sutskever et al., 2014) between DrugChat’s predictions and the ground truth answers in the training data.

3.4 Training and Implementation Details

3.4.1 Data collection and processing

We curated training data for DrugChat from publicly available compound databases, including PubChem, ChEMBL, and DrugBank. The PubChem database¹ contains information on 66,469,244 chemical compounds. We downloaded the bioassay dataset, last updated on January 9, 2025, which includes 4,019,927 unique compounds and 248,667,695 activity records across 636,397 bioassays. The ChEMBL database² provides information on 2,354,965 chemical compounds. We downloaded the SQLite version of the dataset, last updated on February 28, 2023. From this, we selected 3,462 compounds that contain information about administration routes and prodrug status. The DrugBank database³ (version 5.1.10, released on January 4, 2023) contains 16,428 drug entries. We selected 11,583 entries with available SMILES strings, focusing exclusively on small molecules and excluding biotech-classified compounds. After further filtering to retain only entries with annotations for drug indications, pharmacodynamics, or mechanisms of action, we curated 5,846 drug molecules for use in DrugChat. The distribution of drug categories is shown in Fig. 3.10. For each selected compound, we collected its SMILES string along with a variety of properties, such as free-form descriptions (e.g., indications, pharmacodynamics, mechanisms of action) and structured bioassay activities. Additionally, the cytotoxicity, MoleculeNet, and FS-Mol datasets used in our work were curated by Wong et al. (2024), Wu et al. (2018), and Stanley et al. (2021), respectively. After preprocessing and

¹<https://pubchem.ncbi.nlm.nih.gov/>

²<https://www.ebi.ac.uk/chembl/>

³<https://go.drugbank.com/releases/latest>

removal of downstream overlaps described below, the decontaminated PubChem pretraining corpus contained 3.6M unique compounds and 181M activity records across 446K bioassays. DrugBank description, indication, pharmacodynamics, and mechanism-of-action fields were not pretraining targets; PubChem pretraining used only compound–assay activity records.

Using these drugs and their annotated properties, we curated the training data for DrugChat. For each property a of a compound molecule m , we created a triplet consisting of the molecule’s SMILES representation, a textual prompt querying the value of a , and the corresponding ground truth for a . Each property type had its own tailored prompt. For instance, for the property ‘drug indication’, the corresponding prompt is ‘What is its indication?’. The answer is a textual description of the drug compound’s indication, provided by human experts from the DrugBank database. As another example, for the property ‘prodrug status’, the corresponding prompt is ‘Is the molecule a prodrug?’ The ground truth answer is ‘Yes’ if the compound is a prodrug, and ‘No’ otherwise. The distribution of ground truth answers in the cytotoxicity and ChEMBL datasets is shown in Fig. 3.11. We created similar molecule-prompt-answer triplets from the bioassay activities in PubChem and FS-Mol. For each bioassay, we incorporated the textual description of the bioassay into the prompt using the following template: ‘is the compound active in this bioassay: <bioassay description>’. The ground truth answer is ‘Yes’ if the compound is labeled as active in the bioassay, and ‘No’ otherwise. An example prompt for an assay in the FS-Mol dataset is: ‘is the compound active in this bioassay: qHTS assay for identifying a potential treatment of Ataxia-Telangiectasia’. For a complete description of all prompts used across different datasets, please refer to the Prompts subsection below.

3.4.2 Data leakage controls and overlap audit

Compounds in the pretraining and downstream datasets were processed with the same RD-Kit standardization pipeline: cleanup, fragment-parent selection, charge normalization, tautomer canonicalization, and conversion to InChIKey. We matched both the full InChIKey and its first block, which removes stereochemical distinctions. Any pretraining compound matching a down-

stream test compound under either key was excluded. This procedure controls exact duplicates, salts, charge states, tautomers, and stereoisomers and excluded 20K downstream-overlapping compounds.

We separately quantified residual close analogs against the full decontaminated pretraining compound set. Every standardized compound was represented by an ECFP4 fingerprint (Morgan radius 2, 2,048 bits). For each test compound, we exhaustively computed the maximum Tanimoto similarity to any pretraining compound. A compound was classified as having a close analog when its nearest-neighbor similarity was $T \geq 0.7$ and was retained in the close-analog-excluded evaluation when $T < 0.7$. Because exact and standardized variants had already been removed, a compound whose only original match was its own duplicate was not counted as having a close analog.

Table 3.1: Residual close analogs after exact-structure and standardized-variant exclusion. Close analogs have ECFP4 nearest-neighbor Tanimoto similarity $T \geq 0.7$ to the full decontaminated PubChem pretraining compound set.

Test set	Compounds Close analog ($T \geq 0.7$) No close analog		
Cytotoxicity (HepG2/HSkMC/IMR-90)	39,043	33,270 (85.2%)	5,773 (14.8%)
ChEMBL (administration route, prodrug)	3,462	2,109 (60.9%)	1,353 (39.1%)
MoleculeNet—BACE	152	44 (28.9%)	108 (71.1%)
MoleculeNet—BBBP	204	120 (58.8%)	84 (41.2%)
MoleculeNet—ClinTox	148	43 (29.1%)	105 (70.9%)
MoleculeNet—SIDER	143	89 (62.2%)	54 (37.8%)

For close-analog-excluded evaluation, compounds with $T \geq 0.7$ were removed from each test set and Δ AUPRC was recomputed for the remaining compounds. Within each task, all displayed models were evaluated on the same retained compounds for the results in Fig. 3.4 and Fig. 3.5.

We also audited assay, target, and endpoint overlap. The 54 FS-Mol test assays sourced from PubChem were removed before pretraining, although approximately 95% of their targets remained represented through other assays and their binding and inhibition endpoint types were familiar. The HepG2 and IMR-90 cytotoxicity assays were excluded from the PubChem

dataset using a keyword-based filter, and no HSkMC assay was present in PubChem; cytotoxicity involving other cell lines remained common. BACE1 was represented through assays other than the MoleculeNet assay, whereas BBBP had only marginally related permeability assays and ClinTox and SIDER had no direct PubChem bioassay counterpart. Administration route and prodrug status were absent as PubChem bioassay targets.

3.4.3 Model architecture

DrugChat is a multi-modal model that integrates information from three distinct modalities: graphs, images, and text. It consists of a GIN (Xu et al., 2018), a ResNet (He et al., 2016), and an LLM (Chiang et al., 2023). For a given molecule, its SMILES string is converted into both a molecular graph and a molecular image using the RDKit software⁴.

In the molecular graph, nodes correspond to the molecule’s atoms, while edges represent the chemical bonds between them. Each atom is defined by its atom type and chirality, with 120 atom types in total, including a special ‘Unknown’ category for unidentified atoms. Atom chirality is categorized into four types: tetrahedral clockwise, tetrahedral counter-clockwise, unrecognized, and other. These atom type and chirality properties serve as the initial features for each node. Chemical bonds are characterized by their type and direction, with bond types classified as single, double, triple, or aromatic, and bond directions as none, end upright, or end downright. These bond properties are used as the initial features for each edge. All node and edge features are categorical, with each category encoded as a vector with learnable parameters. The molecular graph is input into the GIN to learn a representation vector for the entire graph. The GIN leverages the graph’s connectivity, along with the initial node and edge features, to learn multi-layer representations for each node. Using a neighborhood aggregation approach, the GIN iteratively updates each node’s vector by aggregating information from its neighbors and connecting edges (Kipf and Welling, 2017). After K layers of representation learning, information is propagated through K -hop paths across the graph. An average pooling operation is then applied

⁴<https://www.rdkit.org>

to compute the mean of the final node representations, yielding a single vector that summarizes the entire graph. The GIN in DrugChat consists of five layers and approximately 1.9 million parameters. Both node and edge embeddings have a dimensionality of 300. The GIN was pretrained using a self-supervised context prediction approach (Hu et al., 2020b), in which the model learns to predict a molecule’s subgraphs based on its surrounding subgraphs. Pretraining was performed on 2 million unlabeled molecules from the ZINC15 database (Sterling and Irwin, 2015).

For the molecular image, we employed a ResNet to extract a representation vector. The ResNet processes the input image through multiple layers of 2D convolution, where each layer applies convolutional filters to detect specific patterns in the image or in the feature maps from the previous layer. We used a ResNet-18 (He et al., 2016) model, which consists of 18 convolutional layers and approximately 11 million parameters. A global average pooling layer converts the output of the final convolutional layer into a molecular image representation vector with a dimensionality of 512. Following the approach in ImageMol (Zeng et al., 2022a), the ResNet was pretrained on 10 million unlabeled images of drug-like, bioactive molecules from the PubChem database using self-supervised learning (SSL) techniques, such as molecular image reconstruction and contrastive learning. These SSL methods enable the ResNet to map structurally similar molecules to nearby points in the embedding space, allowing it to learn molecular features at scale without requiring human-annotated labels.

After extracting representation vectors from the molecule using the GIN and ResNet, we apply two separate linear layers — each consisting of a matrix multiplication followed by a bias addition — referred to as adapters, to transform these molecular representations into a format compatible with the LLM. LLMs typically use Transformer decoders (Vaswani et al., 2017b) to model natural language as sequences of tokens, with each token represented as a vector (Brown et al., 2020). In DrugChat, the transformed molecular representations are treated as tokens and appended to the sequence of language tokens derived from the input prompt. This combined sequence is then passed into the LLM, which uses multi-head self-attention (Vaswani et al.,

2017b) to generate new tokens. These generated tokens constitute the final prediction. DrugChat employs Vicuna-13B (Chiang et al., 2023) as its LLM, which has 13 billion parameters. Vicuna-13B was finetuned from LLaMA-13B (Touvron et al., 2023) using a dataset of 70K user-shared conversations from ShareGPT.com (containing interactions between humans and ChatGPT). It retains the architecture of LLaMA-13B, including 40 Transformer layers, 40 attention heads, and an embedding dimension of 5120. The base model was pretrained on a multi-terabyte text corpus — including Wikipedia and various web sources — to predict the next token given preceding context. In our experiments, the LLM weights were frozen and not updated during training. The adapter that converts molecular graph representations into LLM-compatible tokens is a linear layer with input dimension 300 and output dimension 5120, totaling 1.5 million parameters. The adapter for molecular image representations is similarly a linear layer, with an input dimension of 512 and output dimension of 5120, totaling 2.6 million parameters. Because the molecular image is a deterministic 2D rendering of the graph, it adds no chemical information and does not encode 3D geometry. Its role is to provide a representation learned through ImageMol pretraining that differs from the GIN representation of the same graph.

For the parameter-matched graph-only control DC-G-M, we replace the image feature with a second projection of the pooled 300-dimensional graph embedding through a 300×512 linear map. The following trainable projection therefore has the same shape and parameter count as the image projection but receives no additional molecular information. DC-G-M and DC-FS each contain 4,167,680 trainable projection parameters. In DC-FS, 1,541,120 belong to the graph branch and 2,626,560 to the image branch; the ordinary graph-only model contains 1,541,120. These projections constitute the full trainable parameter set in this comparison.

For a target answer T that has L text tokens, DrugChat computes the probability of generating T as follows:

$$p(T \mid M, P) = p_W(T_1 \mid M, P) \prod_{i=2}^L p_W(T_i \mid M, P, T_{<i}), \quad (3.1)$$

where M represents the input molecule and P is the input prompt. We denote the i -th token as T_i and all preceding tokens as $T_{<i}$. Model parameters are denoted by W . The generated token sequence is compared to the ground truth tokens to compute the negative log-likelihood (NLL). The parameters W are optimized by minimizing the sum of NLL over all training data.

3.4.4 Model training

We pretrained DrugChat for one epoch on the decontaminated PubChem molecule–prompt–answer triplets. After applying the compound and assay exclusions described above, each retained compound–assay record and its paired target information was presented at most once. The global batch size of 48 comprised 24 multimodal sequences on each of two GPUs, the largest per-device batch that fit the 13B-parameter LLM, graph and image encoders, and their activations under mixed precision. We used AdamW (Loshchilov and Hutter, 2019a) with a constant learning rate of 10^{-5} , $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay of 0.001. Because approximately 3% of the noisy PubChem bioassay labels were active, inactive and active examples received loss weights of 1 and 30, respectively.

The held-out validation loss decreased from 0.0266 to 0.0031 and had plateaued by the end of pretraining; the training loss was also stable. We selected one epoch because the validation trajectory showed no further improvement and additional passes over noisy labels could promote memorization. This was a resource-limited training run rather than an exhaustive scaling study.

We then fine-tuned DrugChat for five epochs on the MoleculeNet and FS-Mol training splits with batch size 64 and AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 0.01). The learning rate increased linearly from 10^{-5} to 5×10^{-4} during the first epoch and then followed cosine decay to 2×10^{-5} . DrugBank, cytotoxicity, and ChEMBL used nested five-fold scaffold cross-validation: learning rate and epoch count were selected by two-fold cross-validation within the four training folds, after which the model was retrained on those folds and evaluated on the held-out fold. DC, DC-G, and DC-I used the same PubChem pretraining, cross-validation folds, hyperparameters, optimizer schedule, and update count; the only difference was which molecular modality was

provided to the LLM. All experiments used NVIDIA A100 GPUs with 80 GB memory.

3.4.5 Baselines

For DrugBank, GPT-4 received the molecule’s SMILES string and the property question in the following prompt:

```
Given the SMILES of a molecule: {SMILES}, {question}
```

Here, {question} was replaced with the relevant inquiry about indication, pharmacodynamics, mechanism of action, or overview. Galactica-6.7B was fine-tuned with LoRA on each dataset using the same downstream splits as DrugChat. Galactica was selected as the primary molecular-string baseline because its scientific pretraining includes SMILES and its tokenizer provides dedicated SMILES delimiters. Each molecule was supplied in Galactica’s native format:

```
Given the drug [START_I_SMILES]{smiles}[END_I_SMILES], the  
answer to the question '{question}' is
```

The delimiter tokens identify the enclosed sequence as SMILES. Galactica must recover connectivity, branches, and ring closures from character-level or short-substring tokens, whereas DrugChat receives explicit graph connectivity and a separately pretrained image representation. Because the underlying language-model architectures also differ, this is a practical supervised molecular-string baseline rather than a strict representation-only ablation.

For cytotoxicity, ChEMBL, MoleculeNet, and FS-Mol, fine-tuned Galactica used the same decision-score protocol as DrugChat. ChatGLM, FastChat-T5, and LLaMA were retained as reference prompting baselines and received the SMILES and binary question without downstream fine-tuning. For cytotoxicity, the prompt was:

```
Given the SMILES of a molecule: {SMILES}, is the molecule  
cytotoxic to {cell}?
```

where `{cell}` was replaced with HepG2, HSkMC, or IMR-90. For ChEMBL, the binary question asked whether the drug was taken orally, administered parenterally, applied topically, or was a prodrug. MPNN, Chemprop, Text2Mol, KV-PLM, and CLAMP were trained and evaluated through their official codebases.

All cytotoxicity, ChEMBL, and DrugBank baseline values were generated in our runs on the same splits and under the same scoring protocol as DrugChat. Every MoleculeNet baseline was also re-run using the same splits and test compounds, common RDKit canonicalization, the same per-assay Δ AUPRC definition, and the same macro-averaging procedure. On FS-Mol, all models use the same 157 tasks and 56,220 pairs in the held-out tasks split and the same 58 tasks and 16,181 pairs in the held-out tasks and compounds split. Only the FS-Mol KV-PLM and CLAMP values were taken from the literature (Seidl et al., 2023); all other FS-Mol values came from our runs. DrugChat additionally received PubChem bioassay pretraining, whereas fine-tuned Galactica did not; DC-FS controls for this asymmetry.

3.4.6 Prompts

The prompts used by DrugChat to predict cytotoxicity to HepG2, HSkMC, and IMR-90 were ‘Is the molecule cytotoxic to human liver carcinoma cells (HepG2)?’, ‘Is the molecule cytotoxic to primary skeletal muscle cells (HSkMC)?’, and ‘Is the molecule cytotoxic to human lung fibroblast cells (IMR-90)?’. The prompts used by DrugChat on the ChEMBL dataset were: ‘Is the drug taken orally?’, ‘Is the drug administered parenterally?’, ‘Is the drug applied topically?’, and ‘Is the molecule a prodrug?’. The prompts used by DrugChat on the DrugBank dataset were ‘What is its indication?’, ‘What are its pharmacodynamics?’, ‘What is its mechanism of action?’, and ‘What is its overview?’. For the MoleculeNet dataset, DrugChat used the prompt template: ‘Is the compound active in `<task description>`?’ For the FS-Mol dataset, DrugChat used the prompt template: ‘Is the compound active in this bioassay: `<bioassay description>`?’

3.4.7 Model evaluation

We evaluated the free-form predictions of drug indications, pharmacodynamics, and mechanisms of action using both human assessment and automated metrics.

Human evaluation. For each question, evaluators received the drug name, a link to its DrugBank entry, and the DrugBank reference text for the requested field. Answers from the models were presented together in a randomized order without model identities. The evaluator decomposed the reference into distinct biological or chemical facts and identified the one or two facts most important to the question. An answer was scored 2 (correct) when it included the key facts and its remaining content was consistent with established information about the drug; 1 (partially correct) when it included some reference facts but not the complete set of key facts; and 0 (incorrect) otherwise.

The rubric requires a compound- and question-specific correct statement. A generally true but irrelevant statement, or fluent text with no correct reference fact, receives 0. A correct therapeutic class cannot compensate for an incorrect core mechanism; accordingly, the gliclazide mechanism answer in Fig. 3.3 is scored as incorrect.

Automatic evaluation metrics. We conducted automatic evaluations using three metrics: semantic similarity, BLEU (Papineni et al., 2002), and METEOR (Lavie and Denkowski, 2009) scores. Semantic similarity was calculated as the cosine similarity between the sentence embeddings of the ground-truth and model-predicted texts, with embeddings generated using a pretrained sentence Transformer model All-MiniLM-L6-v2 (Reimers and Gurevych, 2019). Let SE represent the sentence embedding model. The embedding of the ground-truth text t_g is denoted as $e_g = \text{SE}(t_g)$, while the embedding of the predicted text t_p is $e_p = \text{SE}(t_p)$. The cosine similarity

between these embeddings is defined as

$$\cos(e_g, e_p) = \frac{\langle e_g, e_p \rangle}{\|e_g\|_2 \|e_p\|_2},$$

where $\langle e_g, e_p \rangle$ represents the dot product between the two vectors, and $\|e_g\|_2$ denotes the L2 norm of the vector e_g . With this sentence Transformer, the embeddings of semantically similar sentences are positioned closer in the embedding space, resulting in higher cosine similarity values. For BLEU scores (Papineni et al., 2002), we used the BLEU-1 score without applying the brevity penalty (Papineni et al., 2002). The BLEU-1 score is a special case of the general BLEU- n metric, which measures the modified precision of n -grams (sequences of n consecutive words). Let $\hat{y}$ represent the predicted sentence, and y the ground-truth sentence. Define $G_n(\hat{y})$ as the set of all n -grams in the predicted sentence. Let $C(s, y)$ be an indicator function, which equals 1 if the n -gram s appears in y ; otherwise, $C(s, y) = 0$. The BLEU- n score is then calculated as follows:

$$\text{BLEU}_n(\hat{y}, y) = \frac{\sum_{s \in G_n(\hat{y})} \min(C(s, \hat{y}), C(s, y))}{\sum_{s \in G_n(\hat{y})} C(s, \hat{y})}.$$

The METEOR score (Lavie and Denkowski, 2009) is viewed as an enhancement over the BLEU score. It is calculated using the harmonic mean of unigram precision and recall, with greater emphasis on recall. Additionally, it accounts for stemming and synonym matching, in addition to exact word matching.

Structural-usability audit. We independently classified every free-form output according to whether it constituted a usable answer. The categories were: placeholder only (only the masked token <compound>), empty (zero-length output), non-prose (chemical-formula-like tokens rather than prose), truncated stub (a lead-in that did not reach a proposition), and abstention (for example, “Unknown”). Their frequencies were 0.87%, 0.51%, 0.29%, 0.22%, and 0.15%, respectively. Factual correctness was assessed separately by the human rubric above.

Multi-turn evaluation. We sampled 50 DrugBank compounds. For each compound, an initial question asked about its description, indication, pharmacodynamics, or mechanism of action, followed by up to three adaptive questions, yielding 148 follow-up turns. Each follow-up answer was scored for context retention (0 or 1), indicating whether it referred to the compound under discussion or information established earlier, and factual correctness (0, 0.5, or 1), indicating incorrect or fabricated, partially correct, or substantially correct content relative to DrugBank and supporting references. We calculated aggregate means and, at the compound level, Spearman associations between the initial-answer score and mean follow-up context retention or correctness.

Structured-task decision score and statistics. At the first generated-token position, we applied the language-model softmax to the output logits and extracted $P(\text{Yes})$ and $P(\text{No})$. The continuous decision score was $s = P(\text{Yes})/P(\text{No})$. Because the softmax covers the full vocabulary, these two probabilities need not sum to one; the ratio is therefore a decision score, not a calibrated probability of correctness. $P(\text{Yes})$ alone and $-P(\text{No})$ gave comparable results, and the ratio was selected because it performed slightly better. We used this score to compute ΔAUPRC , defined as average precision minus the positive rate, for ChEMBL, cytotoxicity, MoleculeNet, and FS-Mol.

Figures report means with error bars as defined in their captions. Baselines were compared with DrugChat using two-sided Welch tests at $p < 0.05$; the standard and close-analog-excluded settings were tested separately. Asterisks indicate significant comparisons, while unmarked comparisons are not statistically significant. For the principal structured-task panels, the sampling units were five cross-validation folds or five repeated runs. With $n = 5$, the smallest attainable two-sided Wilcoxon signed-rank p -value is 0.0625, even when one model is better on all five paired observations, so this rank test cannot identify significance at 0.05. We therefore use mean ΔAUPRC differences as the primary comparison and Welch p -values as supporting information.

3.5 Experiments

3.5.1 Free-form predictions of drug properties

We evaluated DrugChat’s ability to predict key drug properties, including indications, pharmacodynamics, and mechanisms of action. Drug indications refer to the diseases that a compound is intended to treat or its potential applications in clinical trials. Pharmacodynamics describes the biochemical and physiological effects that a compound exerts on human or animal bodies, while the mechanism of action explains how the compound produces these effects. These properties are typically described in free-form texts within the literature. We employed DrugChat to generate free-form textual predictions for these properties using specific prompts: ‘what is its indication?’, ‘what are its pharmacodynamics?’, and ‘what is its mechanism of action?’. We curated these properties from the DrugBank database (Wishart et al., 2006), which includes 5,846 unique drug molecules — both approved and experimental — spanning diverse chemical categories (Methods; Fig. 3.10). Additionally, each drug molecule in DrugBank is accompanied by an expert-written overview that summarizes essential information about the molecule. We assessed the ability of DrugChat to generate overviews for molecules as well. We performed nested five-fold cross-validation with scaffold-based splits (Bemis and Murcko, 1996) to reduce compound-series bias (Zeng et al., 2022b; Yang et al., 2019b; Baumann and Baumann, 2014). Molecules sharing a Murcko scaffold (Bemis and Murcko, 1996) were grouped together and assigned to the same fold, ensuring that the same scaffold did not occur in both the training and held-out subsets. This procedure reduces the risk that performance is driven by closely related compound series. We report the mean and standard deviation across folds (Fig. 3.2a,c). We compared DrugChat with GPT-4 (Achiam et al., 2023), a general-purpose LLM, and Galactica-6.7B (Taylor et al., 2022), a scientific LLM pretrained on scientific text that includes molecular representations. Both baselines received the molecule’s SMILES string in the prompt. Galactica was additionally fine-tuned with LoRA on each DrugBank training fold and received the molecule in its native SMILES-delimiter format (Methods).

Pharmaceutical-science experts evaluated factual accuracy against DrugBank references while blinded to model identity. Answers were randomized and scored as correct (2), partially correct (1), or incorrect (0) using a key-fact rubric described in Methods. DrugChat attained a mean score of 0.949, compared with 0.629 for fine-tuned Galactica and 0.460 for GPT-4 (Fig. 3.2a). The incorrect-response rates for DrugChat were 37.5% for indication, 41.2% for pharmacodynamics, 41.0% for mechanism of action, and 40.9% for overview (Fig. 3.2b). GPT-4 was incorrect on 64.7–70.0% of the same items and fine-tuned Galactica on 36.4–70.0%. Thus, DrugChat improves on the baselines but still requires expert verification. Fig. 3.3 illustrates the scoring rubric.

Besides human evaluation, we used semantic similarity, BLEU (Papineni et al., 2002), and METEOR (Lavie and Denkowski, 2009). Semantic similarity measures alignment between predicted and reference texts using language-model sentence embeddings and ranges from -1 to 1 , with higher values indicating stronger semantic agreement. BLEU and METEOR measure word-level correspondence between predicted and reference texts and range from 0 to 1 , with higher values indicating better agreement. BLEU is defined here as BLEU-1 without a brevity penalty. METEOR incorporates unigram precision and recall, stemming, and synonym matching (Methods). Averaged across indication, pharmacodynamics, mechanism of action, and overview, DrugChat achieved semantic similarity of 0.460 and METEOR of 0.220 , compared with 0.420 and 0.176 for fine-tuned Galactica and 0.317 and 0.141 for GPT-4 (Fig. 3.2c). Fine-tuned Galactica had the highest BLEU (0.324 versus 0.311 for DrugChat and 0.175 for GPT-4), consistent with its shorter outputs: 29.5 words on average, compared with 34.3 for DrugChat and approximately 44 for the reference answers. We consider human ratings as the most important metric in the evaluation of the free-form predictions rather than treating any single automatic metric as sufficient. A separate audit found that 2.04% of DrugChat outputs were structurally unusable: 0.87% contained only the masked compound token, 0.51% were empty, 0.29% contained non-prose chemical-formula-like tokens, 0.22% were truncated stubs, and 0.15% were abstentions. The remaining outputs were well-formed prose, but structural usability does not imply factual

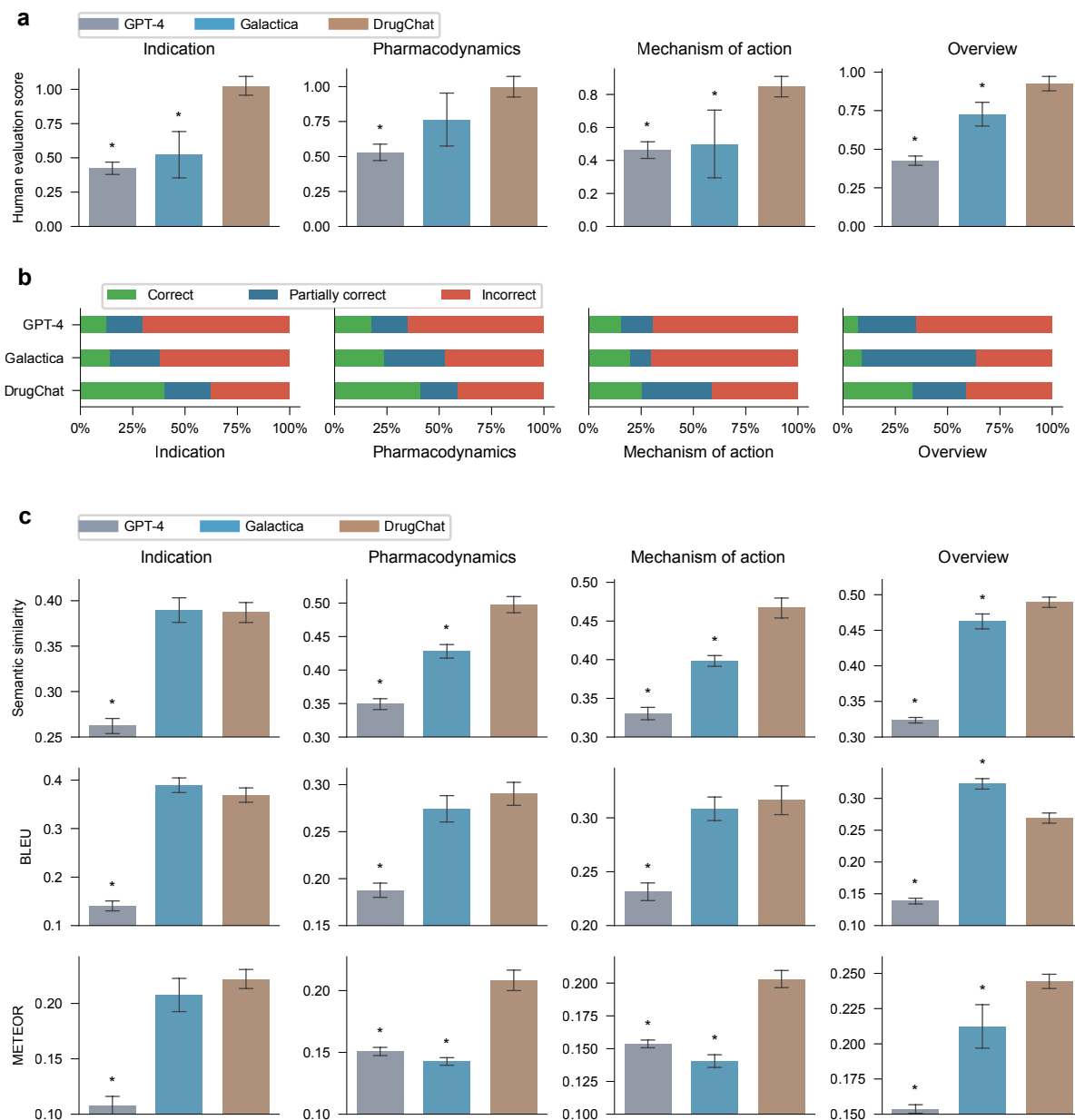


Figure 3.2: DrugChat outperformed GPT-4 and Galactica in predicting drug indications, pharmacodynamics, mechanisms of action, and overviews expressed in free-form texts. a, Mean human-evaluation scores on a 0–2 scale, with results reported as average scores and standard deviations from five-fold cross-validation. b, Proportions of correct, partially correct, and incorrect responses. c, Semantic similarity, BLEU, and METEOR. Galactica was fine-tuned on the same downstream folds as DrugChat. Asterisks mark baselines significantly different from DrugChat by a two-sided Welch test ($p < 0.05$); unmarked comparisons are not statistically significant.

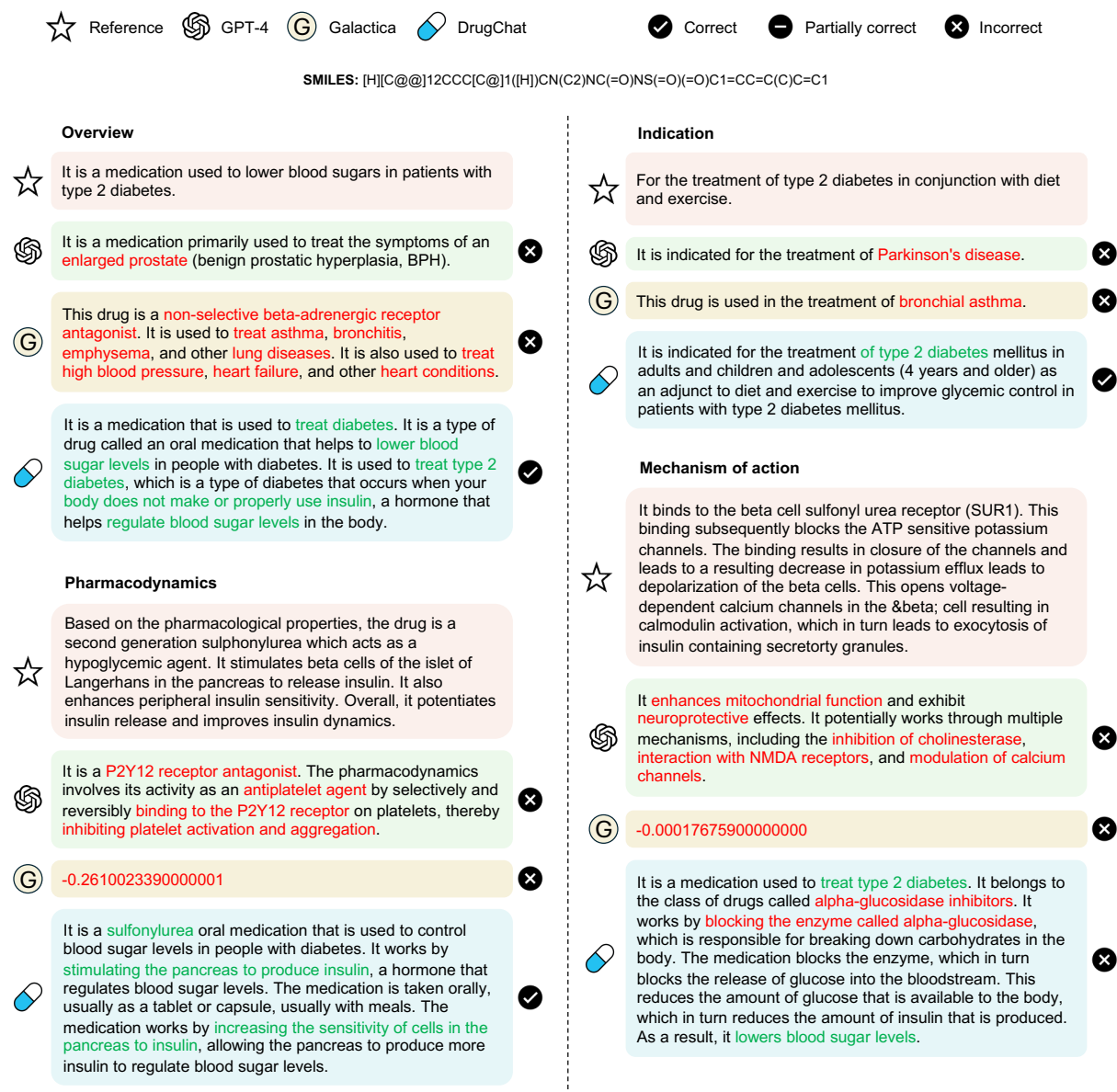


Figure 3.3: Examples of free-form predictions generated by DrugChat, fine-tuned Galactica, and GPT-4, with their human-evaluation scores. Green text indicates correct content and red text indicates incorrect content. DrugChat correctly identifies gliclazide as a treatment for type 2 diabetes and describes its stimulation of pancreatic insulin release.

correctness.

Galactica is a strong string-based baseline because its scientific pretraining includes SMILES and it uses dedicated SMILES delimiters. DrugChat instead supplies an explicit graph representation and a separately pretrained representation of the corresponding 2D depiction. The

two models differ in language-model architecture, so this comparison is not a strict architectural ablation.

3.5.2 Performance on multi-task QSAR modeling

In addition to generating detailed free-form texts, DrugChat performs structured QSAR tasks by generating “Yes” or “No.” We use the ratio of their first-token probabilities as a continuous decision score and report ΔAUPRC , defined as AUPRC minus the proportion of active compounds. Because the expected AUPRC of a random classifier is the positive rate, ΔAUPRC measures improvement over this task-specific baseline (Methods).

We evaluated DrugChat on a cytotoxicity dataset (Wong et al., 2024), the ChEMBL dataset (Gaulton et al., 2012), and the MoleculeNet benchmark (Wu et al., 2018). The cytotoxicity dataset contains 39,043 compounds and 117,129 measurements across three tasks assessing cytotoxicity to HepG2 human liver carcinoma cells, primary skeletal muscle cells (HSkMC), and IMR-90 human lung fibroblasts. HepG2 cells are commonly used to study hepatotoxicity and general cytotoxicity, whereas HSkMC and IMR-90 provide complementary models of toxicity in non-hepatic human cells (Wong et al., 2024). For HepG2, DrugChat was asked, “Is the molecule cytotoxic to human liver carcinoma cells (HepG2)?”; analogous prompts were used for HSkMC and IMR-90 (Methods). Predicting toxicity to human cells is challenging because it requires relating molecular properties to effects in distinct cellular environments.

The ChEMBL dataset comprises 3,462 compounds and 13,847 labels across four tasks: whether a compound can be administered orally, parenterally, or topically, and whether it can act as a prodrug. Administration route depends on physicochemical properties, interactions with biological tissues, and pharmacokinetics. Because a drug may support more than one route—for example, amoxicillin can be administered orally or intravenously—each route was treated as a separate binary task. The fourth task predicts prodrug status. A prodrug is initially inactive or less active and undergoes metabolic or chemical conversion in the body to release its pharmacologically active form.

For MoleculeNet (Wu et al., 2018), we used BACE (one task), BBBP (one task), ClinTox (two tasks), and SIDER (27 tasks). BACE contains binding measurements for 1,522 compounds against human β -secretase 1, a therapeutic target in Alzheimer’s disease (Yan and Vassar, 2014). BBBP contains approximately 2,000 compounds annotated for blood–brain barrier penetration, an important property for drugs targeting the central nervous system. ClinTox contains 1,491 compounds and evaluates clinical-trial toxicity and FDA approval status. SIDER contains 1,427 drugs annotated for side effects across 27 system-organ classes.

For cytotoxicity and ChEMBL, we used nested five-fold cross-validation with scaffold-based splits and report the mean and standard deviation across folds. For MoleculeNet, we followed the benchmark’s standard training, validation, and test splits: BACE and BBBP use scaffold splits, whereas ClinTox and SIDER use random splits. MoleculeNet experiments were repeated five times (Methods).

Baselines included the GNN-based models MPNN (Gilmer et al., 2017) and Chemprop (Yang et al., 2019b; Heid et al., 2023); the multimodal models Text2Mol (Edwards et al., 2021), KV-PLM (Zeng et al., 2022b), and CLAMP (Seidl et al., 2023); and general-purpose prompted LLMs. The GNN models directly output activity scores, whereas the multimodal models combine molecular and text encoders to represent both compounds and task descriptions. The prompted LLMs received the SMILES string and a binary question. Galactica-6.7B (Taylor et al., 2022) was fine-tuned with LoRA on the same downstream splits as DrugChat. Across the 35 paired cytotoxicity and ChEMBL task–fold values, DrugChat achieved mean Δ AUPRC of 0.269, compared with 0.174 for fine-tuned Galactica ($p = 3.7 \times 10^{-5}$; Fig. 3.4). DrugChat’s Δ AUPRC on the four administration-route and prodrug tasks ranged from 0.192 to 0.372 (Fig. 3.4). These labels are curated drug-development annotations rather than PubChem bioassay endpoints. A direct search of 456,658 PubChem assay descriptions found only rare incidental mentions of the relevant administration-route and prodrug terms ($< 0.01\%$), not corresponding assay labels.

After removing test compounds with a close analog in the decontaminated PubChem pretraining set, the mean Δ AUPRC changes across the cytotoxicity and ChEMBL tasks were

−0.069 for DrugChat and −0.074 for CLAMP (Fig. 3.4). The similar changes suggest that much of the decrease reflects the smaller, structurally atypical retained subsets rather than a selective loss for DrugChat.

On MoleculeNet, DrugChat had higher Δ AUPRC than fine-tuned Galactica on BACE (0.270 versus 0.176, $p = 0.0078$) and SIDER (0.083 versus 0.042, $p = 0.0062$), while the models were comparable on BBBP (0.186 versus 0.162, $p = 0.37$) and ClinTox (0.108 versus 0.118, $p = 0.92$; Fig. 3.5). In the close-analog-excluded evaluation, DrugChat’s Δ AUPRC increased on BACE, BBBP, and SIDER. ClinTox was the exception; its smaller retained subset had a different positive rate (Fig. 3.5). Moreover, DrugChat remained as the model with the highest Δ AUPRC on BACE and SIDER.

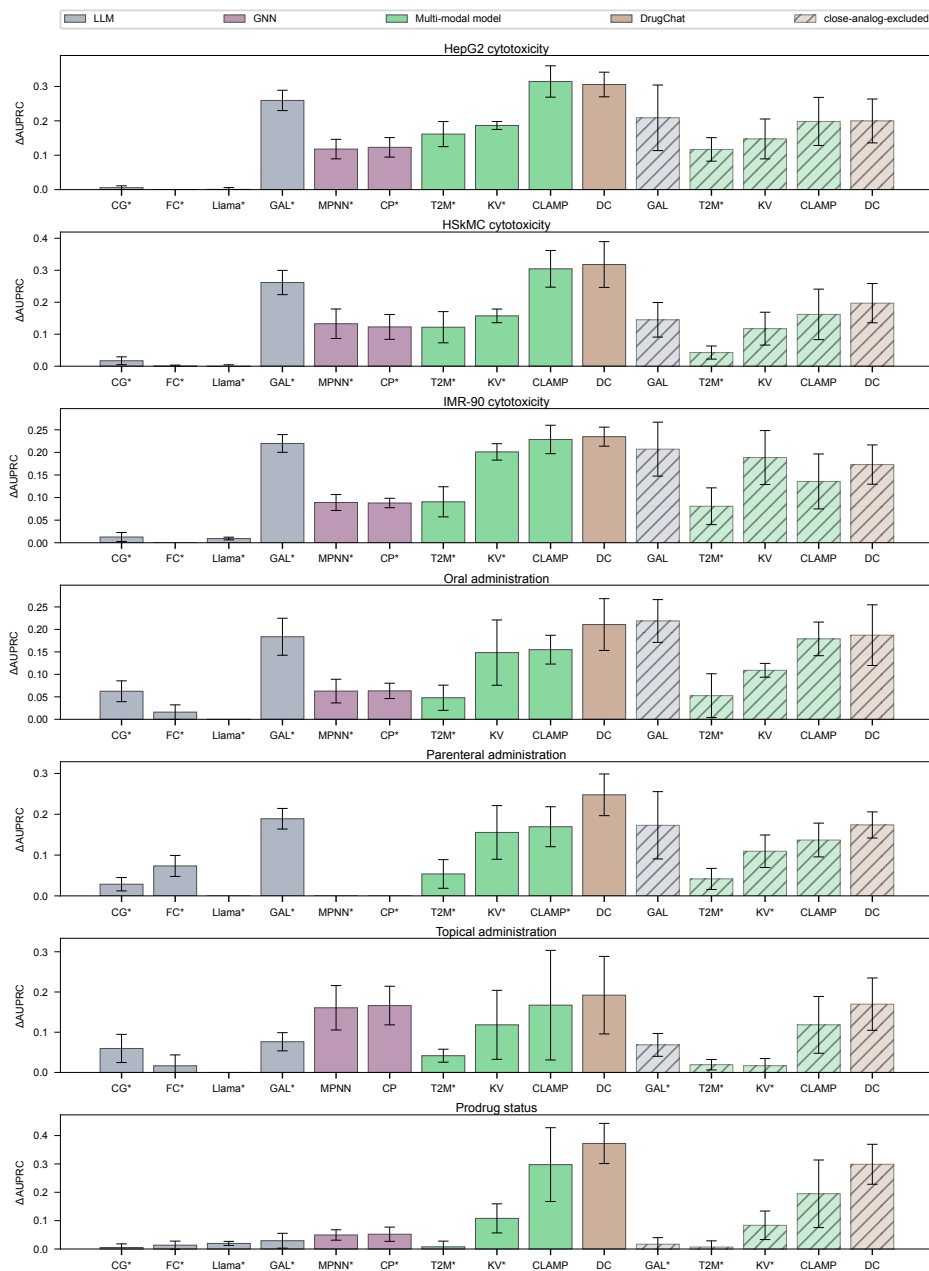


Figure 3.4: Cytotoxicity and ChEMBL structured-task evaluation. Solid bars show the standard ΔAUPRC evaluation for all models. Hatched bars show the close-analog-excluded evaluation for the models with highest performance (fine-tuned Galactica, Text2Mol, KV-PLM, CLAMP, and DrugChat). This evaluation removes test compounds whose nearest ECFP4 neighbor in the decontaminated PubChem pretraining set has Tanimoto similarity ≥ 0.7 . Bars show means and error bars show standard deviations across five cross-validation folds. Asterisks mark baselines significantly different from DrugChat within the corresponding standard or excluded setting by a two-sided Welch test ($p < 0.05$); unmarked comparisons are not statistically significant. GAL, fine-tuned Galactica; CG, ChatGLM; FC, FastChat-T5; CP, Chemprop; T2M, Text2Mol; KV, KV-PLM; DC, DrugChat.

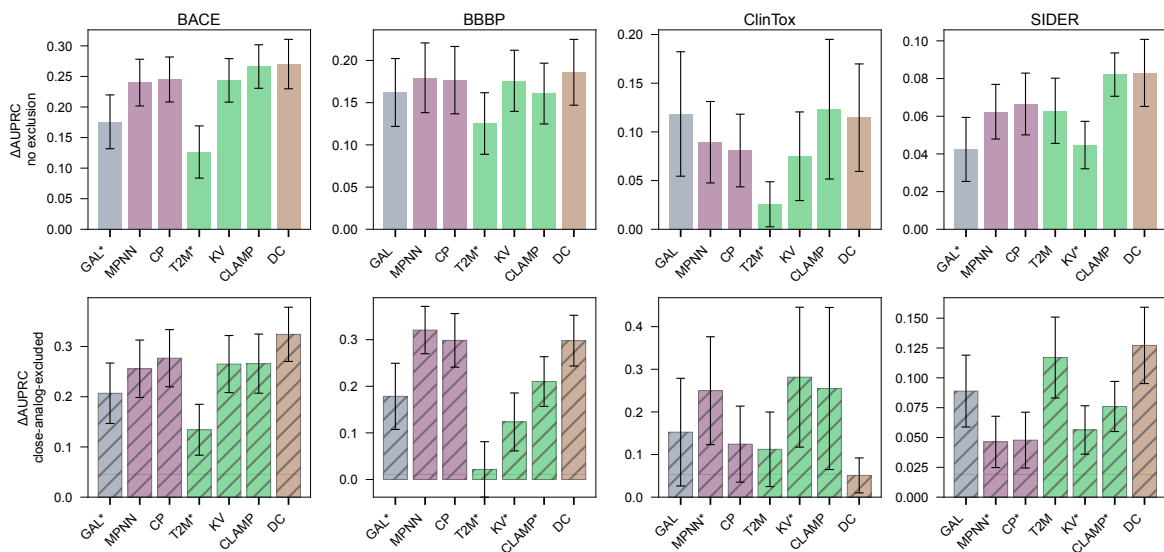


Figure 3.5: MoleculeNet evaluation. The top row shows the standard Δ AUPRC evaluation and the bottom row shows the close-analog-excluded evaluation. Every baseline was re-run on the same splits and test compounds using a common canonicalization, metric, and macro-averaging pipeline. Bars show means and error bars show standard deviations across five runs. Asterisks mark baselines significantly different from DrugChat within the corresponding standard or excluded setting by a two-sided Welch test ($p < 0.05$); unmarked comparisons are not statistically significant. GAL, fine-tuned Galactica; CP, Chemprop; T2M, Text2Mol; KV, KV-PLM; DC, DrugChat.

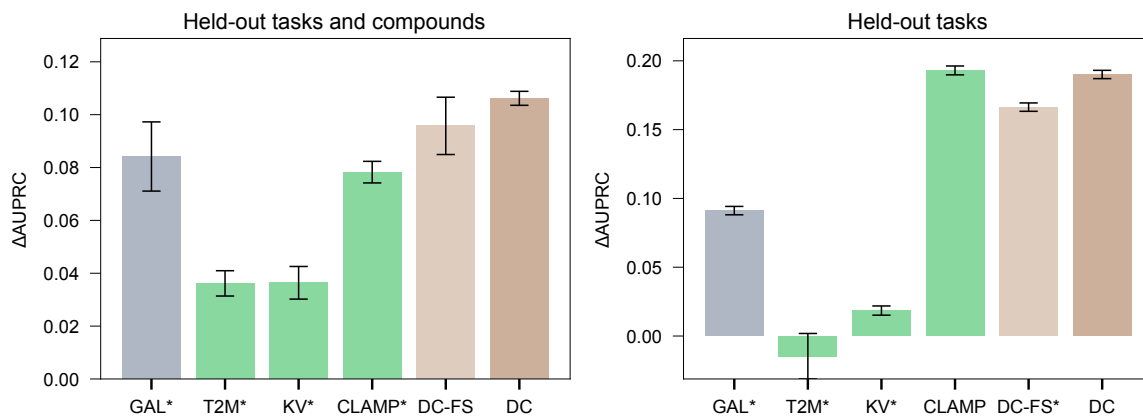


Figure 3.6: Held-out-task generalization on FS-Mol. The left panel is the held-out tasks and compounds split (58 tasks), which additionally excludes compounds appearing in FS-Mol training tasks; the right panel is the held-out tasks split (157 tasks). GAL is Galactica-6.7B fine-tuned on FS-Mol. DC-FS has the same graph–image architecture as DrugChat but omits PubChem bioassay pretraining. Error bars are standard deviations across five runs. Asterisks mark base-lines significantly different from DrugChat by a two-sided Welch test ($p < 0.05$); unmarked comparisons are not statistically significant.

3.5.3 Held-out-task generalization on FS-Mol

FS-Mol (Stanley et al., 2021) contains 5,120 assays and 233,786 unique compounds. We use the official split. The *held-out tasks split* contains 157 assays and 56,220 compound–assay pairs whose tasks are excluded from FS-Mol fine-tuning. The stricter *held-out tasks and compounds split* contains 58 assays and 16,181 pairs and additionally excludes compounds appearing in the FS-Mol training tasks. No test compound–label measurement was a training target, and the pretraining exclusion removed the test compounds from PubChem. The 54 test assays sourced from PubChem were also removed from the PubChem corpus before pretraining, so no FS-Mol test assay was itself a pretraining task. Approximately 95% of test targets nevertheless occur in pretraining through different assays, and binding and inhibition endpoint types are extensively represented.

On the held-out tasks split, DrugChat achieved ΔAUPRC of 0.190, compared with 0.091 for fine-tuned Galactica ($p = 4.3 \times 10^{-11}$) and 0.166 for DC-FS, which omits PubChem bioassay pretraining (Fig. 3.6). On the held-out tasks and compounds split, DrugChat achieved 0.106, com-

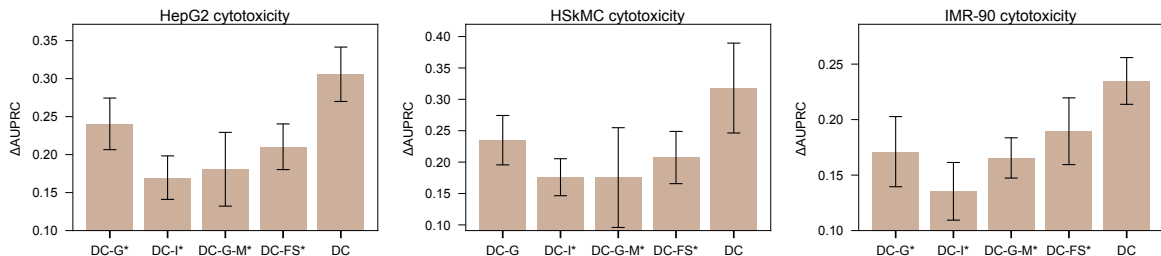


Figure 3.7: Ablation analysis of pretraining and molecular encoders. DC-G and DC-I retain only the graph or image branch, respectively; both use the same PubChem pretraining and downstream fine-tuning protocol as DrugChat (DC). DC-FS has the graph–image architecture but omits PubChem bioassay pretraining. DC-G-M is a graph-only control with the same trainable projection-parameter count as DC-FS. Bars show mean ΔAUPRC and error bars show standard deviations across five cross-validation folds. Asterisks mark variants significantly different from DrugChat by a two-sided Welch test ($p < 0.05$); unmarked comparisons are not statistically significant.

pared with 0.084 for fine-tuned Galactica ($p = 0.018$), 0.096 for DC-FS, and 0.078 for CLAMP (Fig. 3.6). Thus, PubChem pretraining improves performance, but the graph–image architecture retains substantial held-out-task performance without that stage. MPNN and Chemprop are not included because their fixed task-specific heads cannot be applied to held-out assays.

3.5.4 Impact of large-scale pretraining

To assess PubChem bioassay pretraining, we trained DC-FS, the same graph–image architecture initialized without the PubChem checkpoint. On HepG2, HSkMC, and IMR-90, DC-FS achieved mean ΔAUPRC values of 0.210, 0.207, and 0.190, compared with 0.306, 0.318, and 0.235 for DrugChat ($p = 0.0020$, 0.0224, and 0.0275, respectively; Fig. 3.7). Together with the FS-Mol results (Fig. 3.6), this ablation shows that PubChem pretraining improves downstream performance.

Large-scale pretraining exposes the molecular encoders to diverse bioactivity patterns before task-specific fine-tuning, allowing the GIN message-passing layers and ResNet convolutional filters to develop broadly useful feature extractors. The adapters that align molecular representations with textual prompts also benefit from this training, consistent with prior multi-modal

language models (Li et al., 2022, 2023). Starting from these pretrained representations may also reduce overfitting when downstream data are limited (Hendrycks et al., 2019; Zhang et al., 2024).

3.5.5 Advantages of multi-modal integration

To assess the contributions of the molecular encoders, we developed two single-modality variants. DC-G retains only the molecular graph representation produced by the GIN, whereas DC-I retains only the molecular image representation produced by the ResNet. We compared these variants with the full graph–image model on the cytotoxicity tasks (Fig. 3.7). The graph encoder explicitly represents atomic connectivity and bond relationships, while the image encoder supplies a representation learned from large-scale pretraining on standardized molecular depictions. A deterministic 2D molecular depiction contains no chemical information beyond the molecular graph and does not encode 3D geometry. We therefore interpret the ImageMol encoder, pretrained on approximately 10 million depictions, as providing a different learned representation of the same underlying graph rather than additional spatial or 3D information. DC-G and DC-I use the same PubChem pretraining and downstream fine-tuning protocol as DrugChat (Fig. 3.7). To control for trainable parameter count, we added DC-G-M, a graph-only model in which a second 300×512 projection of the pooled graph embedding replaces the image feature (Methods). DC-G-M and DC-FS each have 4,167,680 trainable projection parameters. Because both omit PubChem bioassay pretraining, they isolate the graph–image architecture from that pretraining stage. DC-FS had consistently higher Δ AUPRC than DC-G-M: 0.210 versus 0.181 on HepG2, 0.207 versus 0.175 on HSkMC, and 0.190 versus 0.166 on IMR-90. The corresponding Welch p -values were 0.286, 0.456, and 0.172; the direction was consistent, although no individual comparison reached statistical significance with five folds (Fig. 3.7). The full DrugChat model combines the two pretrained molecular representations before language generation. This design allows the language model to use distinct learned views of the same molecular graph; by contrast, each single-modality variant receives only one of these representations.

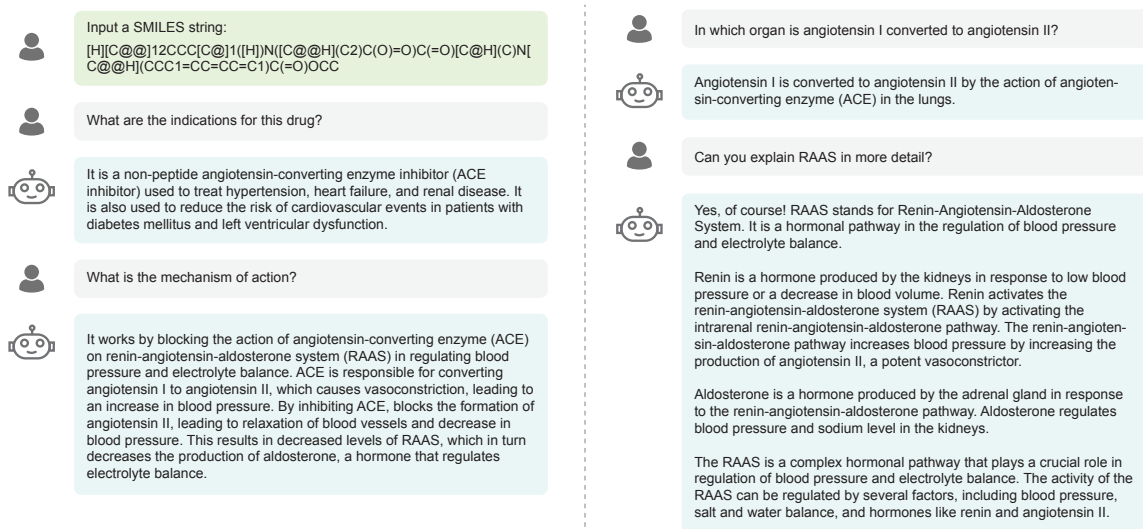


Figure 3.8: An example multi-turn dialogue about ramipril. DrugChat correctly identified the indication and mechanism of ramipril, while maintaining the conversation context across turns.

3.5.6 Interactive exploration of drug properties

DrugChat supports multi-turn interaction about the same molecule. A user can begin with an initial property question and then ask follow-up questions based on the preceding answer, allowing the conversation to move progressively through different aspects of the molecule without resupplying the molecular input. Fig. 3.8 shows an example of multi-turn interaction about ramipril. In this example, DrugChat identifies ramipril as a treatment for hypertension, explains its inhibition of angiotensin-converting enzyme and the conversion of angiotensin I to angiotensin II, identifies the lungs as a major site of this conversion, and provides further explanation of the renin–angiotensin–aldosterone system. The example illustrates how follow-up questions can support a progressively more detailed exploration of a molecule.

We systematically evaluated this capability using 50 DrugBank compounds and 148 adaptive follow-up turns (Methods). DrugChat retained context in 142 of 148 follow-up turns (95.9%), while its mean factual-correctness score was 0.48 on a 0–1 scale (Fig. 3.9). At the compound level, initial-answer correctness was positively associated with mean follow-up correctness (Spearman $\rho = 0.410$, $p = 0.0033$), whereas context retention was not associated with

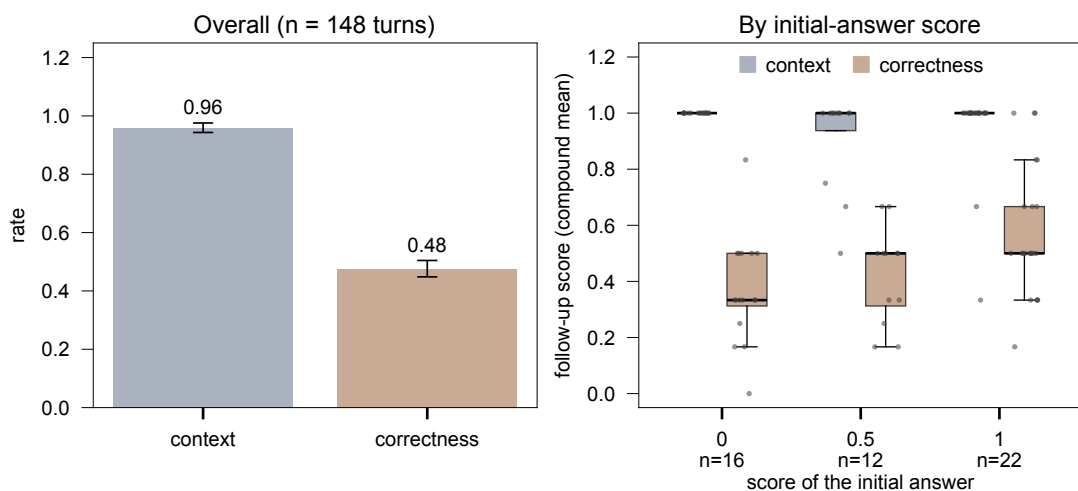


Figure 3.9: Human-scored multi-turn evaluation over 50 compounds and 148 follow-up turns. Left, aggregate context retention and factual correctness; error bars are standard errors of the mean over follow-up turns. Right, compound-level distributions of mean follow-up context and correctness grouped by the score of the initial answer; points are individual compounds. Follow-up correctness is higher after a correct initial answer, whereas context retention remains high across the three groups.

the initial score ($\rho = -0.099$, $p = 0.55$; Fig. 3.9). A correct initial answer therefore provides a useful signal for placing greater confidence in follow-up answers, although it does not guarantee their correctness.

The practical benefit is that a user can request elaboration, redirect the topic, or refine a question without resupplying the molecule. Safety-relevant answers should nevertheless be checked against authoritative sources or results of biomedical experiments.

3.6 Discussion

We present DrugChat, a multi-modal large language model with three principal capabilities. First, DrugChat generates free-form textual predictions for complex drug properties such as indications, pharmacodynamics, and mechanisms of action. Prior models have generally represented these properties using structured outputs such as categorical labels, numerical values, or dose–response curves (Yang et al., 2019a; Lu et al., 2021; Chen et al., 2020; Ota

and Yamashita, 2022; Zeng et al., 2019). DrugChat instead frames the prediction in natural language, allowing richer descriptions that more closely resemble biomedical literature and expert annotations. DrugChat also unifies this free-form molecular question answering with structured QSAR prediction. Its explicit graph and pretrained image representations differ from the SMILES-only representation used by fine-tuned Galactica. The comparison is task dependent: the graph-image model shows its clearest advantages in held-out-task transfer, prodrug prediction, and human-rated free-form generation, while the string baseline is competitive on several individual endpoints and has a higher BLEU score on DrugBank (Figs. 3.2–3.6).

Second, DrugChat supports prompt-based multi-task learning over an extensible task space. Unlike conventional multi-task models with fixed, predefined outputs (Dahl et al., 2014; Yang et al., 2019b; Jastrzębski et al., 2016), DrugChat uses natural-language prompts and an instruction-tuned language model. Users can therefore specify biological or pharmacological questions without adding task-specific output heads, as would be required for traditional multi-task GNNs (Yang et al., 2019b; Hu et al., 2020b) or MLPs (Dahl et al., 2014; Unterthiner et al., 2014; Ramsundar et al., 2015). The same framework handles traditional QSAR tasks, including toxicity and bioactivity prediction, as well as free-form questions about drug indications, pharmacodynamics, and mechanisms of action. Changing the prompt switches the task for the same molecule (Fig. 3.1a). Multi-modal models such as KV-PLM (Zeng et al., 2022b), CLAMP (Seidl et al., 2023), and Text2Mol (Edwards et al., 2021) can address structured classification or retrieval tasks, whereas DrugChat also generates free-form answers. The FS-Mol results demonstrate generalization to tasks held out from downstream fine-tuning, rather than to wholly novel biological tasks. None of the 157 test assays was used as a PubChem pretraining task, and the held-out tasks and compounds split additionally removes compounds appearing in FS-Mol training tasks. However, approximately 95% of the test targets are represented in PubChem through other assays, and binding and inhibition endpoint types are familiar. Generalization to genuinely novel protein targets therefore remains an open evaluation setting. The no-PubChem DC-FS control retains substantial performance and exceeds the published baselines on the held-out tasks and compounds

split (Fig. 3.6).

Third, DrugChat supports multi-turn interaction, allowing users to refine questions and explore drug properties through contextual dialogue. This capability complements the single-pass predictions produced by KV-PLM, CLAMP, Text2Mol, and Galactica. Our multi-turn evaluation separates contextual dialogue from factual reliability. DrugChat retained context in 95.9% of follow-up turns, but its mean follow-up correctness was 0.48; a correct initial answer was positively associated with later correctness (Fig. 3.9). Context retention therefore supports iterative questioning, but conversational coherence is not evidence that an answer is factually correct. Because context retention already left little headroom and single-turn DrugBank correctness was higher than for GPT-4 and fine-tuned Galactica, we did not add a separate multi-turn baseline comparison.

In DrugChat, molecular representations of structured properties, such as cytotoxicity and bioassay activity, and unstructured properties, such as drug indications, pharmacodynamics, and mechanisms of action, are embedded within a shared latent space. This unified representation space enables the model to learn diverse and complementary features, potentially improving its generalization capability. By extracting and integrating information across different types of molecular knowledge, DrugChat can uncover patterns and relationships that might be missed by models trained in isolated domains. This cross-domain learning framework supports prediction across both structured and unstructured tasks. The graph and image encoders also provide distinct learned views of the same molecular structure. The image modality should nevertheless be interpreted narrowly. A deterministic 2D depiction is graph-equivalent and does not represent 3D geometry. The parameter-matched DC-G-M control shows that adding an equally sized graph projection does not reproduce the consistent Δ AUPRC advantage of DC-FS over DC-G-M, although the individual comparisons are not statistically significant with five folds (Fig. 3.7). The evidence therefore supports the value of a differently pretrained representation of the molecular graph, rather than additional spatial or 3D chemical information.

Although DrugChat is primarily designed for analyzing candidate molecules with un-

known properties, it also offers potential benefits for known drugs. Curated databases such as DrugBank and PubChem provide authoritative, human-verified information, but they are inherently static and limited to existing annotations. DrugChat can generate hypotheses about previously unrecognized properties of known compounds, such as alternative therapeutic indications, pharmacodynamic effects, or mechanisms of action. This may be particularly useful for drug repositioning, which seeks to identify new uses for approved drugs; such hypotheses would require expert review and experimental validation.

Although DrugChat processes molecular structure as input, it does not merely function as a chemical-similarity predictor. To assess its ability to generalize beyond structural similarity, we adopted scaffold-based splits for the DrugBank, cytotoxicity, and ChEMBL datasets, ensuring that compounds in the held-out fold did not share Murcko scaffolds with compounds in the corresponding training folds. This setup reduces the potential for information leakage through scaffold-level overlap. The overlap audit further argues against direct memorization of molecular structures. Exact structures, salts, charge states, tautomers, and stereoisomers matching downstream compounds were removed before pretraining, and performance remained competitive after excluding test compounds with ECFP4 similarity ≥ 0.7 to the decontaminated pretraining set (Figs. 3.4 and 3.5). The PubChem corpus contains 3,617,028 compounds, while close analogs associated with these benchmarks form only a small subset. Pretraining used one epoch, so each compound–assay record and its paired target information was encountered at most once. Together, these observations support the interpretation that pretraining improves general molecular representations and related biomedical knowledge rather than memorizing a specific subset of compounds or tasks.

The broader reliability limitations and future directions raised by DrugChat, including factual verification, confidence calibration, interpretability, broader training data, integration with simulation tools, and evaluation on genuinely novel targets and expert workflows, are discussed in Chapter 5.

3.7 Data and Code Availability

The datasets curated and used in this work are available at <https://huggingface.co/datasets/youweiliang/drugchat>

3.7.1 Code Availability

The source code of this work is available at <https://github.com/youweiliang/drugchat>

3.8 Additional Dataset Diagnostics

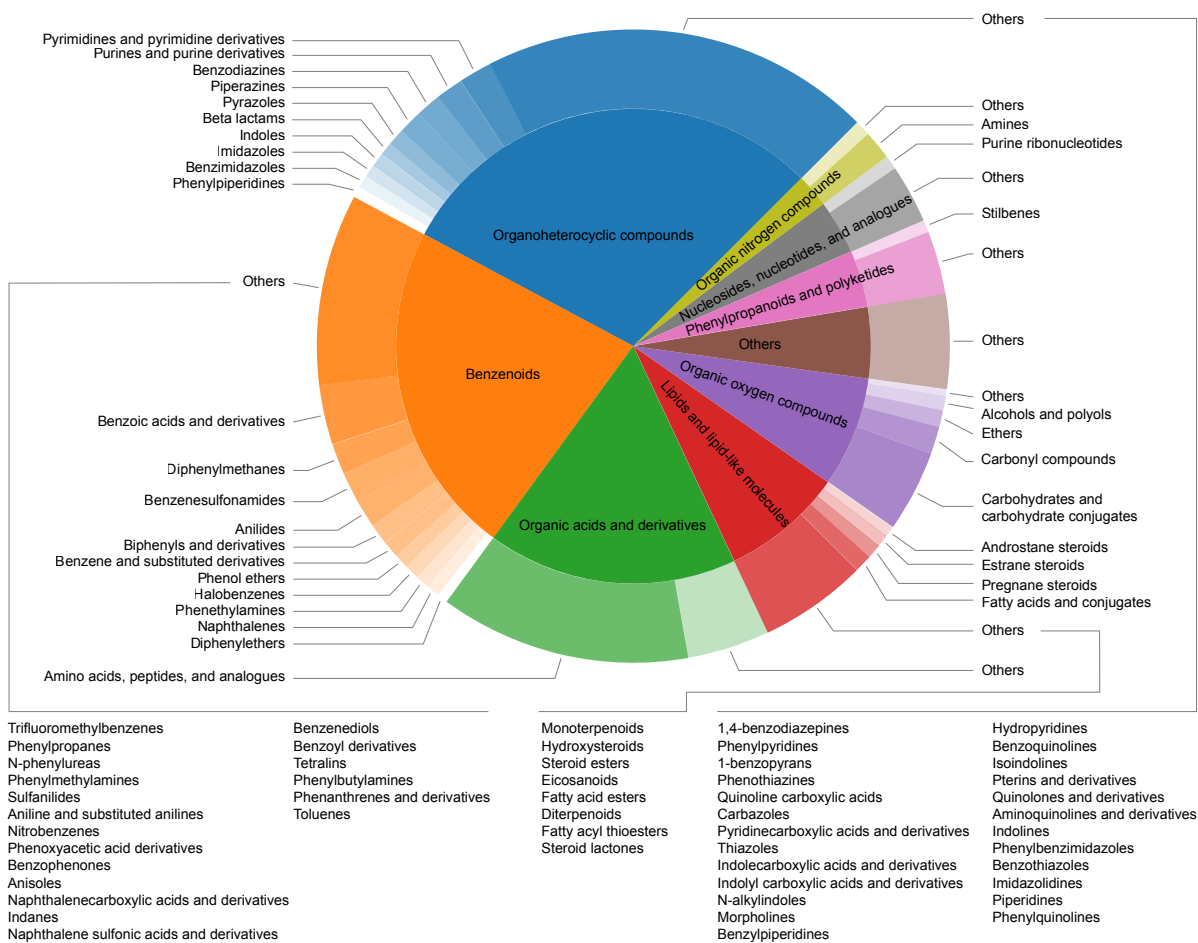


Figure 3.10: Our curated dataset features a diverse distribution of compound categories. The inner disk represents the compound superclasses, while the outer ring shows the corresponding subclasses.

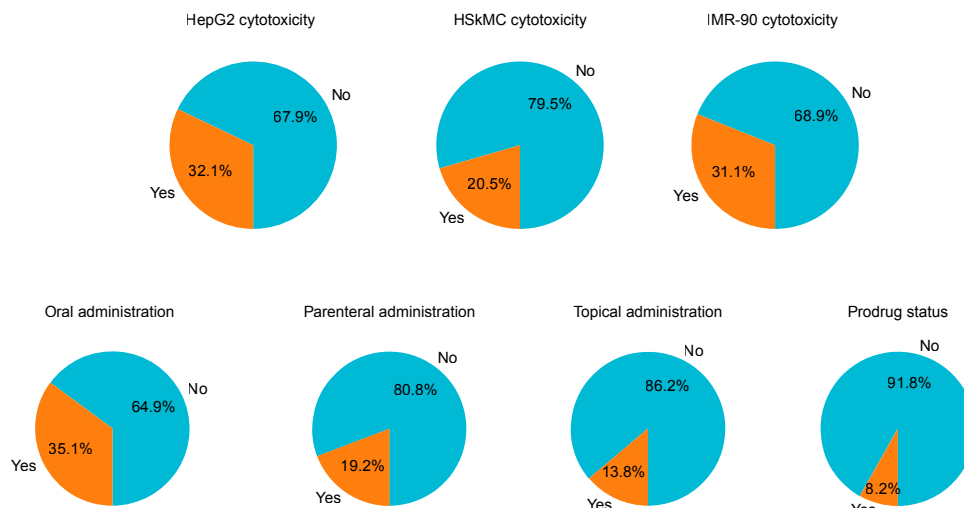


Figure 3.11: The distribution of ground-truth answers in the test set across the tasks of predicting cytotoxicity, administration routes, and prodrug status.

3.9 From Adaptive Prediction to Adaptive Supervision

DrugChat shows how adaptive multimodal models can flexibly answer different molecular questions by integrating graph, image, and language representations. In this setting, adaptation occurs primarily at the level of task specification and representation. However, many scientific imaging problems introduce a further challenge: the supervision itself is unreliable. In cryo-electron tomography, labels are expensive to obtain, often noisy, and severely imbalanced across macromolecular species. The next chapter therefore shifts from adapting inputs and predictions to adapting the training signal itself.

Chapter 3, in full, has been submitted for publication of the material as it appears in “A Multi-Modal Large Language Model for Free-Form, Open-Ended, and Interactive Prediction of Properties and Mechanisms of Candidate Drug Molecules” by Youwei Liang, Ruiyi Zhang, Zinnia Ma, Digvijay Singh, Yongce Li, Mingjia Huo, Chengzhan Gao, Hamidreza Rahmani, Satvik Bandi, Li Zhang, Robert Weinreb, Atul Malhotra, Danielle A. Grotjahn, Linda Awdishu, Trey Ideker, Michael Gilson, and Pengtao Xie. The dissertation author was the primary researcher and author of this paper.

Chapter 4

RobPicker: Adaptive Supervision for Cryo-ET Particle Picking

4.1 Chapter Overview

This chapter presents RobPicker as adaptive supervision for noisy and imbalanced scientific imaging. EViT adapts which input tokens are processed, and DrugChat adapts how molecular prediction tasks are specified. RobPicker moves adaptation to the training signal itself by learning a label correction network and a data reweighting network through validation-guided meta-learning.

4.2 Motivation

Cryo-electron tomography (cryo-ET) has revolutionized structural biology by enabling high-resolution visualization and structure determination of macromolecules in their near-native states within the complex cellular environment in situ (Beck and Baumeister, 2016; Mahamid et al., 2016; McCafferty et al., 2024; Nogales and Mahamid, 2024; Cai et al., 2018; Huang et al., 2022; Pyle et al., 2025). Accurately identifying and localizing macromolecules in cryo-ET data is crucial for understanding their functions, interactions, and dynamics within the cellular context (Lučić et al., 2013). With advances in instrumentation (Berger et al., 2023; Eisenstein et al., 2023), data processing (Pyle and Zanetti, 2021; Watson and Bartesaghi, 2024), and automation (Chen et al., 2019; Eisenstein et al., 2024), the volume of in-cell cryo-ET data is rapidly expanding in public

repositories such as EMPIAR (Iudin et al., 2016). By training neural networks on annotated tomograms, recent deep learning approaches have shown promise in automatically identifying macromolecular complexes (referred to as particles) from cryo-ET data (Gao et al., 2021; Zeng et al., 2021; Lamm et al., 2022; Zeng et al., 2023; Moebel et al., 2021; de Teresa-Trueba et al., 2023; Liu et al., 2024a). In particular, macromolecule segmentation networks such as 3D U-Net (Çiçek et al., 2016) have been successfully applied to segment and pick particles, such as ribosomes (Moebel et al., 2021; de Teresa-Trueba et al., 2023). However, these networks often struggle with robustness due to two key challenges: noise in training labels introduced by human or automated annotations, and imbalanced distributions across macromolecule species (Moebel and Kervrann, 2022; Rice et al., 2023).

The low signal-to-noise ratio (SNR) of cryo-ET raw data, combined with incomplete angular sampling due to the physical constraints of the microscope (i.e., the “missing wedge” artifact (Yan et al., 2019)), presents significant challenges for detecting and classifying particles within the crowded cellular milieu, particularly those of smaller size (Turk and Baumeister, 2020). Recent cryo-ET denoising methods (Bepler et al., 2020; Liu et al., 2022; Zhang et al., 2023) have improved the SNR of tomograms; however, they cannot fully eliminate noise (Wiedemann and Heckel, 2024). Noisy tomograms also complicate particle annotation which is necessary for training supervised deep learning models for particle picking. Typical annotation strategies include manual, semi-automated, and fully automated approaches. In manual annotation, human experts browse tomograms and mark particle locations. While manual annotation can be accurate for large particles, it is extremely labor-intensive (Chen et al., 2017). To accelerate the process, semi-automated and automated pipelines have been developed using template matching (Böhm et al., 2000) and deep learning models (Chen et al., 2017; Wagner and Raunser, 2020; de Teresa-Trueba et al., 2023). However, template matching often produces false positives (Cruz-León et al., 2024), and supervised deep learning models struggle to identify small particles, especially when training data are scarce (Ermel et al., 2024). In summary, particle annotation in cryo-ET is inherently difficult—particularly for small particles—and time-consuming. Automated

annotations frequently introduce inconsistent or erroneous labels (referred to as noisy labels). When such noisy labels are used to train deep learning models, the resulting models fail to accurately identify particles in previously unseen tomograms. Recent semi-supervised and self-supervised learning approaches (Rice et al., 2023; Huang et al., 2024; Stojanovska et al., 2025) have been proposed to reduce the reliance on clean annotations required by supervised methods. However, these approaches often struggle to generalize to experimental tomograms, which exhibit substantial variation in noise statistics, missing-wedge artifacts, acquisition conditions, and cellular context. Such variability can lead to discrepancies between the self-supervised pre-training stage and downstream particle-picking performance (Bandyopadhyay et al., 2022; Harastani et al., 2025).

Training macromolecule segmentation networks is further impeded by the significant imbalance in the occurrence of different macromolecular species within the cellular environment (Luo et al., 2019). Larger, more abundant species occupy more voxels in tomograms and tend to exhibit clearer boundaries against the background, whereas smaller or less abundant species—though often playing vital roles in cellular processes—are underrepresented and occupy fewer voxels. Because training macromolecule segmentation networks typically relies on voxel-wise loss functions—such as Dice loss (Zhao et al., 2020; Yeung et al., 2022)—to update model parameters, larger and more abundant species exert a greater influence on model updating (Li et al., 2020b). This imbalance results in skewed performance: models become proficient at identifying large, well-delineated species but struggle to detect smaller, rarer ones. The inability to consistently detect small particles in cryo-ET data has limited the understanding of smaller macromolecules and their critical roles in biological systems.

To address these limitations, we introduce RobPicker, a novel meta-learning (Hospedales et al., 2021) framework that improves the robustness of macromolecule segmentation networks for cryo-ET. Unlike traditional supervised deep learning methods, which rely heavily on large quantities of well-labeled data, RobPicker is designed to overcome challenges posed by labeling noise and imbalanced species distributions. It achieves this by automatically learning a label

correction network to rectify labeling errors and a data reweighting network to place greater emphasis on smaller and rarer macromolecule species. The learning of these auxiliary networks is guided by evaluating the performance of the macromolecule segmentation network on a validation set with cleaner labels and a more balanced species distribution. The corrected and reweighted data are then used to train a more robust macromolecule segmentation network (Fig. 4.1a).

Through comprehensive evaluations on experimental cryo-ET datasets and benchmarks, RobPicker demonstrates significantly improved robustness in identifying particles across various cell types, even when labels are noisy and species distributions are imbalanced. Moreover, we demonstrated the effectiveness and generalizability of RobPicker by applying it to subtomogram averaging for protein structure determination across multiple species.

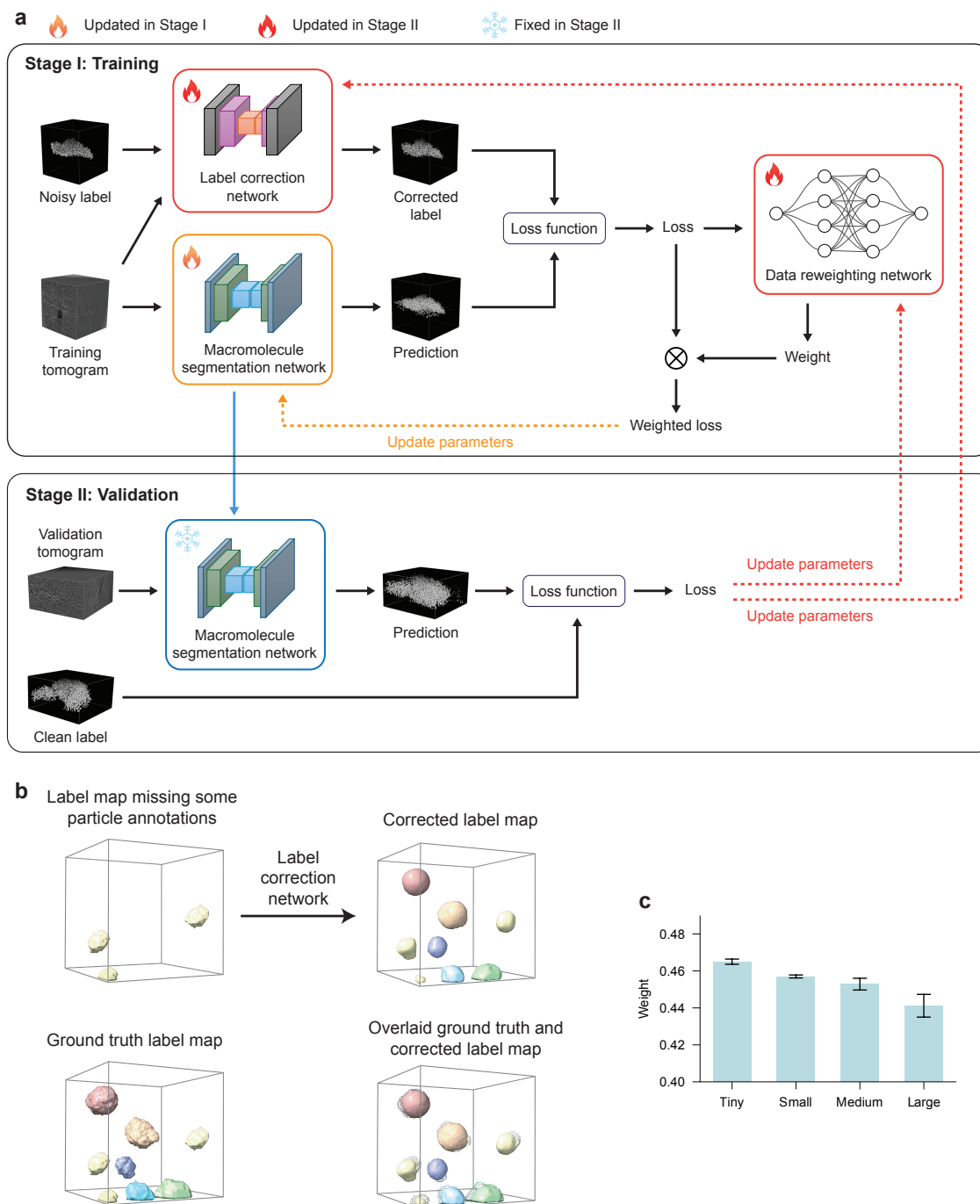


Figure 4.1: Overview of RobPicker. a, RobPicker jointly trains macromolecule segmentation, label correction, and data reweighting networks through bi-level optimization. Stage I updates the segmentation network using corrected labels and reweighted training losses; Stage II updates the auxiliary networks using loss on clean, balanced validation data. b, Label correction recovers missing particle annotations; meshes show ground truth, solid objects show corrected labels, and colors denote species. c, Data reweighting assigns larger weights (mean $\pm$ standard deviation) to smaller species; the x-axis denotes particle size.

4.3 Method Overview

4.3.1 RobPicker overview

RobPicker is composed of three deep neural networks: a macromolecule segmentation network, a label correction network, and a data reweighting network. The macromolecule segmentation network takes a cryo-electron subtomogram (i.e., a 3D patch extracted from a full tomogram) as input and outputs a detection result in the form of a segmentation map, which indicates the likelihood of each macromolecule species at each voxel in the subtomogram. Each subtomogram/patch typically contains one or a few particles. The segmentation map is then used to derive the particle species and their centers through post-processing (Methods). We employed a 3D U-Net (Çiçek et al., 2016) as the macromolecule segmentation network, following prior deep learning pickers (Moebel et al., 2021).

The label correction network is designed to correct noisy labels in the training data. It is a 3D U-Net (Çiçek et al., 2016) that takes a subtomogram and the corresponding particle segmentation map as inputs and produces a corrected segmentation map (Fig. 4.1a and Methods). In practice, the label correction network can share the model backbone with the segmentation network to reduce the number of parameters (Methods). The parameters of the label correction network are updated to minimize the segmentation loss on a separate, smaller dataset (referred to as the validation set) with labels that are cleaner than those in the training set (Zheng et al., 2021) (Stage II in Fig. 4.1a). Such a cleaner validation dataset can be constructed either by retrospectively selecting a subset of tomograms with higher-quality annotations or by curating annotations through multiple experts (Methods). Intuitively, a poorly trained label correction network produces corrupted supervision signals for the segmentation network during the training stage (Stage I in Fig. 4.1a), which leads to degraded segmentation performance in the validation stage and thus a high validation loss, since the validation labels are cleaner. Therefore, minimizing the validation loss can update the label correction network in a direction that enables it to generate improved supervision signals, i.e., cleaner labels. An analysis of the label correction network

trained on the SHREC 2019 cryo-ET dataset (Gubins et al., 2020) revealed that the network recovered missing ground-truth particle labels. An exemplar corrected label map is shown in Fig. 4.1b, underscoring the effectiveness of the label correction mechanism in RobPicker.

In addition to the label correction network, the data reweighting network (Shu et al., 2019) predicts a weight for each input subtomogram to increase the influence of smaller or less abundant particle species on the parameter updates of the segmentation network. Each training subtomogram/patch receives its own weight, and subtomograms/patches containing smaller or rarer particle species are assigned higher weights during training. Specifically, the reweighting network takes as input the prediction loss of a subtomogram and outputs a scalar weight between 0 and 1. This weight is multiplied by the prediction loss, thereby controlling the influence of that subtomogram on model parameter updates—higher weights lead to greater influence (Stage I in Fig. 4.1a). The reweighting network is implemented as a multi-layer perceptron (MLP) with a single hidden layer. Similar to the label correction network, the reweighting network is updated by minimizing the validation loss on the validation set, where balanced data sampling is applied to increase the frequency of less abundant species (Methods). Analysis on the SHREC 2019 cryo-ET dataset (Gubins et al., 2020) shows that the trained reweighting network assigns larger weights to subtomograms predominantly containing smaller particle species (Fig. 4.1c), supporting the effectiveness of the reweighting mechanism in RobPicker.

4.4 Bi-Level Optimization

4.4.1 The RobPicker framework

RobPicker is formulated as a meta-learning problem with a bi-level optimization (BLO) formulation. A BLO problem consists of two levels of optimization: a lower level and an upper level. In RobPicker, training of the macromolecule segmentation network occurs at the lower level, while training of the DR and LC networks occurs at the upper level. The two levels are nested and mutually dependent: the optimal parameters of the segmentation network from the

lower level are used to define the objective function at the upper level, whereas the non-optimal parameters of the DR and LC networks from the upper level are used to define the objective function at the lower level. Due to this mutual dependency, the model parameters at both levels are updated iteratively, with the detailed procedure described below.

Given a training dataset $D_{\text{tr}} = \{(t_i, a_i)\}_{i=1}^{N_{\text{tr}}}$, where t_i represents an input tomogram, a_i denotes the corresponding macromolecule label, and N_{tr} is the number of training samples, we input t_i into the macromolecule segmentation network $f(t_i; I)$, where I denotes the network weight parameters, to obtain a prediction. Simultaneously, a_i is fed into the LC network $g(a_i; C)$, where C represents its weight parameters, producing a corrected label. A loss function $\ell(f(t_i; I), g(a_i; C))$, defined as the Dice loss (Dice, 1945), is computed to quantify the discrepancy between the prediction and the corrected label. The resulting loss is then passed to the DR network to produce a scalar weight $h(\ell(f(t_i; I), g(a_i; C)); R)$, where the DR network is parameterized by R . The loss is multiplied by this weight. Let $L(D_{\text{tr}}, I, C, R)$ denote the sum of the reweighted losses over the entire training dataset:

$$L(D_{\text{tr}}, I, C, R) = \sum_{i=1}^{N_{\text{tr}}} h(\ell(f(t_i; I), g(a_i; C)); R) \ell(f(t_i; I), g(a_i; C)). \quad (4.1)$$

We train the segmentation network by updating its parameters I to minimize the loss $L(D_{\text{tr}}, I, C, R)$ while keeping the parameters of the LC and DR networks fixed. This leads to the following optimization problem:

$$I^*(C, R) = \operatorname{argmin}_I L(D_{\text{tr}}, I, C, R). \quad (4.2)$$

Here, $I^*(C, R)$ indicates that the optimal trained parameter I^* is a function of the parameters of the LC and DR network, C and R , since I^* is determined by the loss function, which itself depends on C and R . Note that C and R are not optimized in this step, as doing so could lead to a trivial solution where the weight $h(\ell(f(t_i; I), g(a_i; C)); R)$ becomes zero for every i .

Next, we evaluate the trained segmentation network $I^*(C, R)$ on a clean validation dataset

$D_{\text{val}} = \{(t_j, a_j)\}_{j=1}^{N_{\text{val}}}$, where N_{val} denotes the number of validation examples. For each validation tomogram t_j , the segmentation network is used to detect macromolecules, and the resulting prediction $f(t_j; I^*(C, R))$ is compared with the corresponding ground-truth label a_j , yielding a loss $\ell(f(t_j; I^*(C, R)), a_j)$. Let $L(D_{\text{val}}, I^*(C, R))$ denote the total loss over the entire validation dataset:

$$L(D_{\text{val}}, I^*(C, R)) = \sum_{j=1}^{N_{\text{val}}} \ell(f(t_j; I^*(C, R)), a_j). \quad (4.3)$$

We optimize the parameters C and R of the LC and DR networks by minimizing the validation loss, leading to the following optimization problem:

$$\min_{C, R} L(D_{\text{val}}, I^*(C, R)) \quad (4.4)$$

Combining Equations 4.2 and 4.4, we obtain the following bi-level optimization problem:

$$\begin{aligned} \min_{C, R} L(D_{\text{val}}, I^*(C, R)) \\ \text{s.t.} \quad I^*(C, R) = \operatorname{argmin}_I L(D_{\text{tr}}, I, C, R) \end{aligned} \quad (4.5)$$

The two levels of optimization problems are interdependent. The solution from the lower level, $I^*(C, R)$, is used to define the loss function at the upper level, while the optimization variables C and R from the upper level are also involved in defining the loss function at the lower level.

4.4.2 Optimization algorithm

We solve the optimization problem in Equation 4.5 using a hypergradient-based approach. Specifically, we approximate the optimal solution $I^*(C, R)$ through a single-step gradient descent update as follows:

$$I^*(C, R) \approx I'(C, R) = I - \eta_I \nabla_I L(D_{\text{tr}}, I, C, R). \quad (4.6)$$

where η_I denotes the learning rate.

We substitute the approximation $I^*(C, R) \approx I'(C, R)$ into the upper-level loss function,

yielding an approximate loss $L(D_{\text{val}}, I'(C, R))$. We then update C and R using gradient descent with respect to the approximate loss:

$$C \leftarrow C - \eta_c \nabla_C L(D_{\text{val}}, I'(C, R)), \quad (4.7)$$

$$R \leftarrow R - \eta_r \nabla_R L(D_{\text{val}}, I'(C, R)), \quad (4.8)$$

where η_c and η_r denote the respective learning rate. The gradient $\nabla_C L(D_{\text{val}}, I'(C, R))$ can be computed using the chain rule:

$$\nabla_C L(D_{\text{val}}, I'(C, R)) = \frac{\partial I'}{\partial C} \frac{\partial L(D_{\text{val}}, I'(C, R))}{\partial I'}, \quad (4.9)$$

where

$$\frac{\partial I'}{\partial C} = -\eta_I \nabla_{C,I}^2 L(D_{\text{tr}}, I, C, R). \quad (4.10)$$

Similarly, the gradient $\nabla_R L(D_{\text{val}}, I'(C, R))$ is computed as:

$$-\eta_R \nabla_{R,I}^2 L(D_{\text{tr}}, I, C, R) \frac{\partial L(D_{\text{val}}, I'(C, R))}{\partial I'}. \quad (4.11)$$

The updates in Equations 4.6, 4.7, and 4.8 are performed iteratively until convergence. The steps of the algorithm are outlined in Algorithm 2.

In Equation 4.11, computing the matrix $\frac{\partial I'}{\partial C} = -\eta_I \nabla_{C,I}^2 L(D_{\text{tr}}, I, C, R)$ and performing multiplication with the vector $\frac{\partial L(D_{\text{val}}, I'(C, R))}{\partial I'} = \nabla_{I'} L(D_{\text{val}}, I'(C, R))$ incur noticeable computational cost. To mitigate this, we adopted the finite difference approximation method (Liu et al., 2019). This method is a numerical technique for estimating derivatives when their analytical forms are either complex or infeasible to compute. By evaluating the function at perturbed points, finite difference approximation provides an efficient way to approximate derivatives. Let I^+ and I^- denote $I + \varepsilon \nabla_{I'} L(D_{\text{val}}, I'(C, R))$ and $I - \varepsilon \nabla_{I'} L(D_{\text{val}}, I'(C, R))$, respectively, where ε is a small

scalar. Using this approach, the matrix-vector multiplication can be approximated as:

$$\begin{aligned} \nabla_{C,I}^2 L(D_{\text{tr}}, I, C, R) \nabla_{I'} L(D_{\text{val}}, I'(C, R)) \approx \\ \frac{\nabla_C L(D_{\text{tr}}, I^+, C, R) - \nabla_C L(D_{\text{tr}}, I^-, C, R)}{2\varepsilon} \end{aligned} \quad (4.12)$$

Similarly, $\nabla_{R,I}^2 L(D_{\text{tr}}, I, C, R) \nabla_{I'} L(D_{\text{val}}, I'(C, R))$ can be approximated as:

$$\begin{aligned} \nabla_{R,I}^2 L(D_{\text{tr}}, I, C, R) \nabla_{I'} L(D_{\text{val}}, I'(C, R)) \approx \\ \frac{\nabla_R L(D_{\text{tr}}, I^+, C, R) - \nabla_R L(D_{\text{tr}}, I^-, C, R)}{2\varepsilon}. \end{aligned} \quad (4.13)$$

This bi-level optimization algorithm was implemented using our Python library Betty (Choe et al., 2023), which also provides the implementation of RobPicker.

Algorithm 2: Bi-level optimization for RobPicker

Input: Training set D_{tr} , validation set D_{val} ;
initial parameters for segmentation network $I^{(0)}$, label correction network $C^{(0)}$, data
reweighting network $R^{(0)}$;
learning rates η_I , η_C , η_R ; maximum iterations T

Output: Learned parameters I , C , R

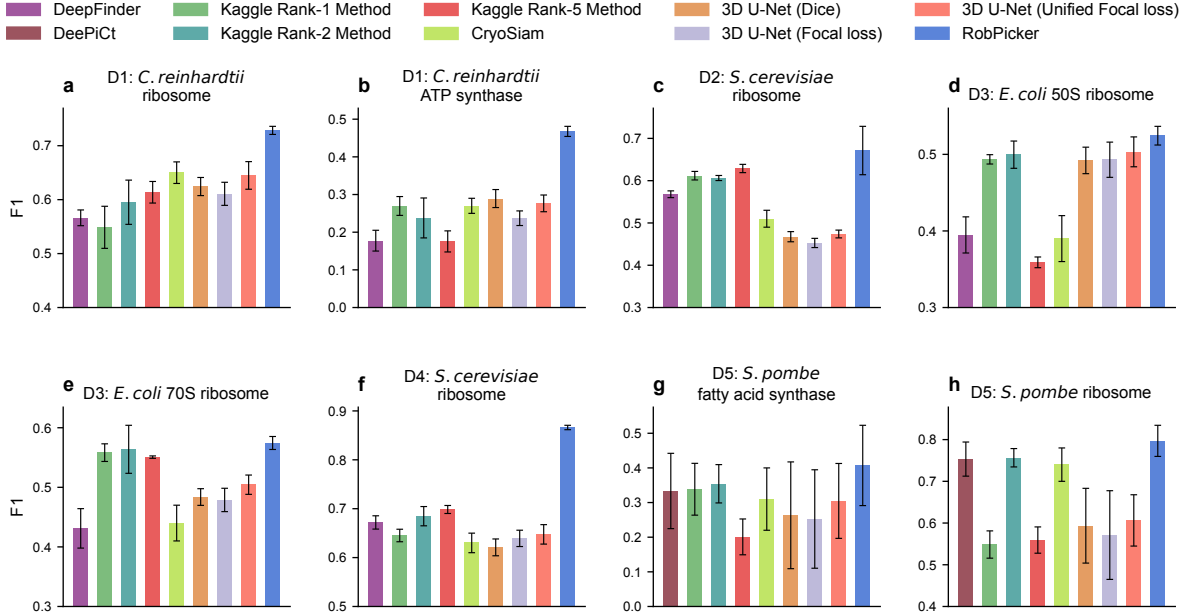
```

1 for  $t \leftarrow 1$  to  $T$  do
    // Approximate inner optimum  $I^*(C, R)$  by a single-step
    // gradient descent
2    $I' \leftarrow I^{(t-1)} - \eta_I \nabla_I L(D_{\text{tr}}, I^{(t-1)}, C^{(t-1)}, R^{(t-1)});$ 
    // Update outer parameters  $C$  and  $R$  using
    // hypergradient
    // The hypergradient is approximated by
    // Equations 4.12 and 4.13
3    $C^{(t)} \leftarrow C^{(t-1)} - \eta_C \nabla_C L(D_{\text{val}}, I');$ 
4    $R^{(t)} \leftarrow R^{(t-1)} - \eta_R \nabla_R L(D_{\text{val}}, I');$ 
5    $I^{(t)} \leftarrow I'$ 
6 return  $I^{(T)}, C^{(T)}, R^{(T)}$ 

```

4.5 Experiments

4.5.1 RobPicker tolerates noise and class imbalance in cryo-ET particle picking



i									
Dataset	Object	Train	Val	Test	Dataset	Object	Train	Val	Test
D1	tomogram	14	2	2	D3	tomogram	26	1	1
	ribosome	5,286	1,087	730		50S	6,013	443	238
	ATP synthase	7,790	751	935		70S	6,318	512	285
D2	tomogram	4	1	1	D4	tomogram	3	1	1
	ribosome	2,986	707	843		ribosome	6,467	1,238	1,496

Figure 4.2: RobPicker performance across real cryo-ET datasets. RobPicker achieved higher mean F1 scores than (1) the supervised methods DeepFinder and DeePiCt; (2) three top-ranked methods from a Kaggle competition for cryo-ET particle picking (Kaggle Rank-1, Rank-2, and Rank-5); (3) three class-imbalance strategies, including class-reweighted Dice loss, Focal Loss, and Unified Focal Loss (Yeung et al., 2022); and (4) the self-supervised method CryoSiam across datasets D1–D5. Panels a–h show the mean and standard deviation. Panel i reports the numbers of tomograms and particles in the training, validation, and test splits.

We conducted experiments on five real cryo-ET datasets, denoted as D1–D5, comprising tomograms with macromolecular annotations across different cell types. The first dataset (D1) contains annotations for 80S ribosomes and ATP synthase in *C. reinhardtii* cells (Kelley et al., 2026) (detailed statistics are provided in Fig. 4.2i). D2 contains annotations for 80S ribosomes

in *S. cerevisiae* cells (Chang et al., 2025). D3 contains annotations for 50S large subunits and fully assembled 70S ribosomes in *E. coli* cells. D4 also includes annotations for ribosomes in *S. cerevisiae* cells. D5 comprises 10 *S. pombe* tomograms with annotations of 731 fatty acid synthase (FAS) complexes and 25,311 ribosomes, resulting in a highly imbalanced species distribution. Details of data annotation are provided in the Methods section.

We compared RobPicker against nine baselines, categorized into four groups (Methods). The first group consists of two established supervised deep learning methods, DeepFinder (Moebel et al., 2021) and DeePiCt (de Teresa-Trueba et al., 2023), which use 3D U-Net with 3D convolutional layers (Çiçek et al., 2016) for multi-class semantic segmentation to pick particles in tomograms. We used the same 3D U-Net architecture in RobPicker as in DeepFinder and DeePiCt to demonstrate that the improved robustness of RobPicker results from the meta-learning framework rather than other factors. The second group consists of three stronger and more recent baselines that achieved top rankings in a Kaggle competition for cryo-ET particle picking. Specifically, we compared RobPicker with the methods that achieved first (Henkel and Khvedchenya, 2025) and second place (Luo and Wang, 2025), for which code and model configurations are publicly available, as well as with the DeepFinder-derived method that placed fifth (Ouertani et al., 2025). We refer to these methods as Kaggle Rank-1, Rank-2, and Rank-5, respectively. All three methods employ ensembles consisting of multiple segmentation or detection networks (Methods). The third group consists of three well-established strategies for handling class imbalance: (1) voxel-wise Dice loss with class reweighting based on species frequency, (2) Focal loss (Lin et al., 2017), which upweights “hard” examples (i.e., those with high classification loss), and (3) Unified Focal Loss (Yeung et al., 2022) with class-balanced reweighting. To ensure a fair comparison, all three baselines employ the same 3D U-Net architecture as RobPicker for particle picking (Methods). The fourth group includes CryoSiam (Stojanovska et al., 2025), a recent self-supervised particle-picking framework pretrained on automatically generated simulated cryo-ET data without manual annotations. The pretrained CryoSiam segmentation network was adapted to each dataset and evaluated using the same protocols as RobPicker (Methods).

For experiments on D1–D5, we used a standard 3D U-Net (Çiçek et al., 2016) as the segmentation backbone. To report model performance, we used the F1 score as the evaluation metric, which ranges from 0 (lowest performance) to 1 (highest performance). A particle prediction is considered a true positive only if its coordinate lies within a predefined distance threshold of a ground-truth annotation of the same species (Methods). The experiments on D1–D4 were repeated three times, and the mean and standard deviation are reported in Fig. 4.2a–f. For D5, we followed the protocol of DeePiCt to use five-fold cross-validation to report the mean and standard deviation of the F1 scores across the five held-out folds (de Teresa-Trueba et al., 2023).

Across D1–D5, RobPicker consistently achieved higher mean F1 scores than the supervised deep learning methods DeepFinder and DeePiCt, with the largest improvements observed for small and rare particle species (Fig. 4.2). For example, for *C. reinhardtii* ATP synthase in D1, RobPicker achieved a mean F1 score of 0.47, significantly outperforming DeepFinder, which achieved a mean F1 score of 0.18—a 163% improvement ($P < 0.001$, two-sided t-test). In addition to ATP synthase, *S. pombe* FAS in D5 is particularly challenging to identify due to its sporadic presence and smaller size (compared to ribosomes) in cells (de Teresa-Trueba et al., 2023), resulting in a highly imbalanced species distribution. By mitigating class imbalance and correcting noisy labels, RobPicker increased the mean F1 score for FAS from 0.33 (DeePiCt) to 0.41, representing a 22.1% improvement. Beyond small particles, RobPicker remained competitive with or superior to DeepFinder on ribosomal targets, achieving substantial improvements on *S. cerevisiae* ribosomes in D2 (0.67 vs. 0.57) and D4 (0.87 vs. 0.67), as well as on *E. coli* 70S ribosomes in D3 (0.57 vs. 0.50). These experimental results indicate that RobPicker robustly tolerated label noise in the training set while requiring only a small number of clean labels for the validation set. In contrast, DeepFinder and DeePiCt exhibited reduced robustness when trained on noisy labels generated using sphere-based particle masks.

On D1–D5, RobPicker also outperformed the three top-ranked Kaggle methods, including Kaggle Rank-1, Rank-2, and Rank-5 (Fig. 4.2). For example, RobPicker achieved F1 scores of 0.73 for ribosomes and 0.47 for ATP synthase on D1, exceeding the Kaggle meth-

ods' scores of 0.55/0.60/0.61 for ribosomes ($P < 0.027$) and 0.27/0.24/0.18 for ATP synthase ($P < 0.013$). On D4 (*S. cerevisiae* ribosomes), RobPicker achieved an F1 score of 0.87 compared with 0.65/0.68/0.70 for the Kaggle methods ($P < 0.003$). These results are notable because the Kaggle Rank-1, Rank-2, and Rank-5 methods use model ensembles, whereas RobPicker uses a single segmentation network.

RobPicker significantly outperformed all three data reweighting baselines on D1–D5 (Fig. 4.2). For example, on the challenging D1 ATP synthase particle-picking task, RobPicker achieved an F1 score of 0.47, compared with 0.29 (class-reweighted Dice, $P = 0.001$), 0.24 (Focal, $P < 0.001$), and 0.28 (Unified Focal, $P < 0.001$), yielding substantially larger relative gains of 61.6%, 97.2%, and 69.1%. For *C. reinhardtii* ribosomes in D1, RobPicker achieved a mean F1 score of 0.73, outperforming class-reweighted Dice (0.62; $P = 0.003$), Focal (0.61; $P = 0.006$), and Unified Focal (0.64; $P = 0.023$), corresponding to relative gains of 16.7%, 19.3%, and 12.9%, respectively. On D2 (*S. cerevisiae* ribosome), RobPicker achieved an F1 score of 0.67, exceeding the performance of class-reweighted Dice (0.47; $P = 0.021$), Focal (0.45; $P = 0.019$), and Unified Focal (0.47; $P = 0.024$), with relative gains of 43.6%, 48.3%, and 41.7%. For *S. cerevisiae* ribosomes in D4, RobPicker achieved an F1 score of 0.87, outperforming class-reweighted Dice loss (0.62; $P < 0.001$), Focal loss (0.64; $P < 0.001$), and Unified Focal loss (0.65; $P = 0.002$).

RobPicker consistently achieved higher particle-picking performance than CryoSiam across D1–D5 (Fig. 4.2). For example, on challenging particles such as *C. reinhardtii* ATP synthase (D1), RobPicker achieved a mean F1 score of 0.47 compared to CryoSiam's 0.27 (72.5% relative improvement; $P < 0.001$). Beyond challenging particles, RobPicker also maintained clear advantages on larger complexes, including *S. cerevisiae* ribosomes in D4 (0.87 vs. 0.62; $P = 0.001$) and D2 (0.67 vs. 0.51; $P = 0.029$), as well as *E. coli* ribosomes in D3 (50S: 0.52 vs. 0.39 with $P = 0.008$; 70S: 0.57 vs. 0.44 with $P = 0.009$).

Collectively, these results demonstrate that RobPicker consistently achieves the highest particle-picking accuracy across D1–D5. It outperforms established supervised methods

(DeepFinder and DeePiCt), recent top-ranked Kaggle competition pipelines (Rank-1, Rank-2, and Rank-5), class-imbalance-aware loss baselines (class-reweighted Dice, Focal Loss, and Unified Focal Loss), and the self-supervised framework CryoSiam. The performance gains are particularly pronounced for challenging particle species where annotation noise and class imbalance are most severe, while RobPicker remains competitive or superior on larger and more abundant ribosomal targets. Notably, these improvements are achieved without relying on model ensembles. Together, the results indicate that RobPicker’s meta-learning framework—integrating validation-guided data reweighting and label correction—provides systematic and robust advantages across diverse experimental conditions, especially in regimes characterized by noisy supervision and rare particle classes.

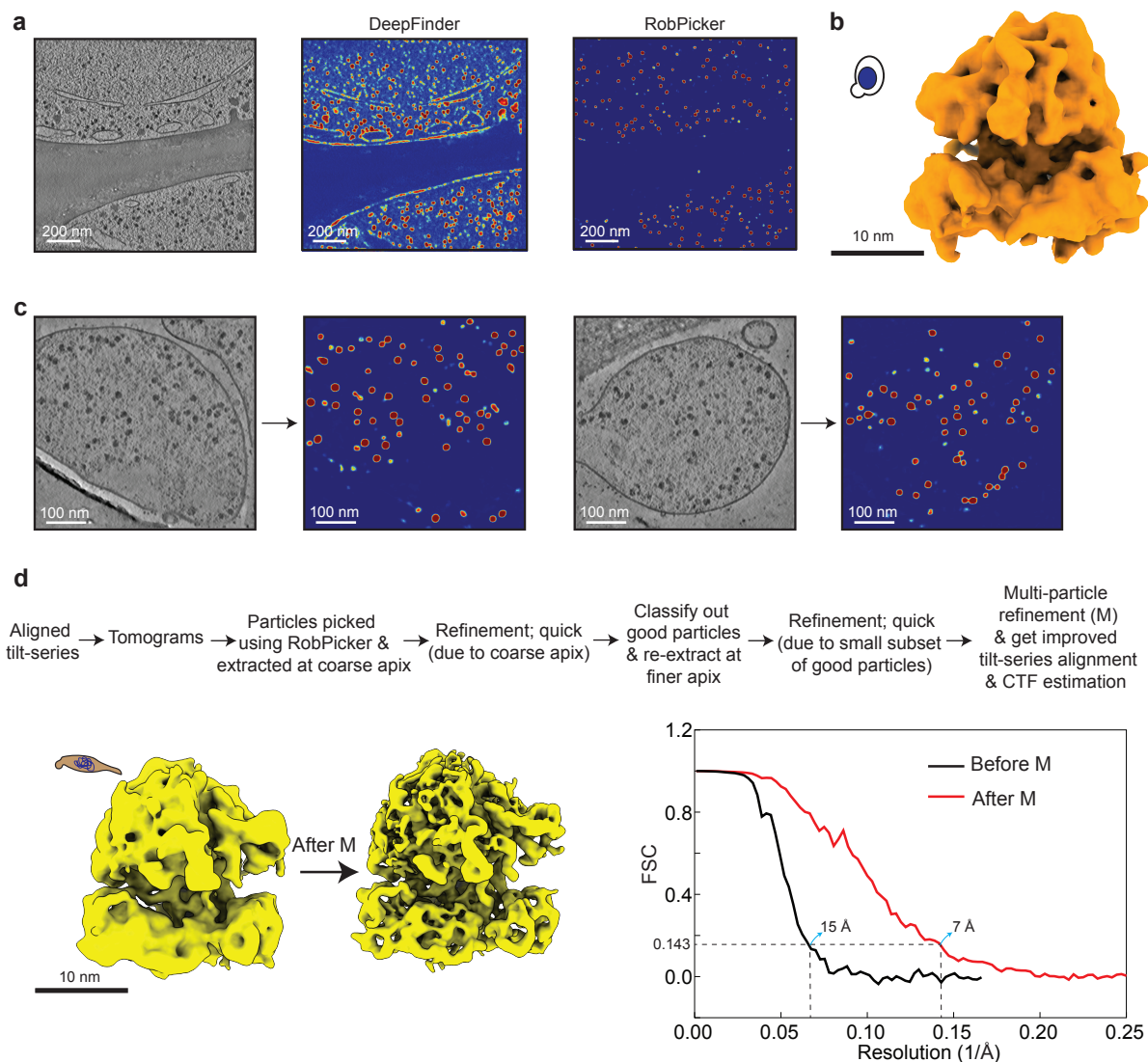


Figure 4.3: RobPicker shows high accuracy and strong cross-organism generalizability in particle picking. **a**, Tomographic slice (first column) of a representative *S. cerevisiae* tomogram, alongside heatmaps of ribosome localization probabilities predicted by DeepFinder and RobPicker. RobPicker exhibits markedly higher specificity, with substantially fewer false positives. **b**, Rapid RELION refinement using RobPicker's picks generated this representative *S. cerevisiae* ribosome map. **c**, The same RobPicker model trained on *S. cerevisiae* tomograms was applied, without retraining, to detect ribosomes in in-cell bacterial tomograms. Tomographic slices (first and third columns) from two representative bacterial tomograms and the corresponding RobPicker prediction heatmaps (second and fourth columns) are shown, highlighting RobPicker's accuracy and specificity in cross-organism particle detection. **d**, An efficient RELION classification and refinement pipeline produced a high-quality bacterial ribosome map (left map), demonstrating RobPicker's strong generalizability across organisms. An additional multi-particle refinement workflow (top panel) further improved the resolution of the ribosome reconstruction (right map). The corresponding FSC curves are shown in the right panel.

4.5.2 RobPicker enables generalizable particle picking across organisms

We performed an experiment to evaluate whether a RobPicker model trained on tomograms from one organism can generalize to another organism without retraining. RobPicker was first trained on four *S. cerevisiae* tomograms from D4 to detect ribosomes. On a held-out *S. cerevisiae* tomogram, RobPicker achieved a substantially higher F1 score than DeepFinder (0.87 vs. 0.67; Fig. 4.2f). As shown in Fig. 4.3a, RobPicker produced more accurate particle probability maps with markedly fewer false positives, indicating improved specificity. We further applied the trained RobPicker model to eight additional held-out *S. cerevisiae* tomograms to detect ribosomes. The detected particles were subjected to rapid 3D refinement followed by 3D classification in RELION (Burt et al., 2024), yielding a representative *S. cerevisiae* ribosome map at approximately 19 Å resolution (Fig. 4.3b).

To assess cross-organism generalization, we applied the same RobPicker model—without retraining—to 14 in-cell tomograms of a different organism, *M. pneumoniae* (bacterial), acquired at a coarse pixel size of 20 Å/pixel, to detect ribosomes. Heatmaps of the predicted ribosome localization probabilities highlighted RobPicker’s accuracy and specificity in cross-organism ribosome detection (Fig. 4.3c). To evaluate picking quality, we performed 3D refinement in RELION using the detected particles, which was computationally efficient due to the coarse pixel size. Subsequent 3D classification yielded a subset of high-quality particles, which were re-extracted at finer pixel size for further refinement. This workflow produced a representative bacterial ribosome map at approximately 15 Å resolution (Fig. 4.3d). Despite substantial differences between the *S. cerevisiae* and *M. pneumoniae* tomograms in cellular context and imaging conditions, the resulting high-quality bacterial ribosome map demonstrates that RobPicker generalizes effectively to organisms not included in training.

We further applied the multi-particle refinement tool “M” (Tegunov et al., 2021) to jointly optimize tilt-series alignment and contrast transfer function (CTF) parameters across the full tomograms (Fig. 4.3d). Application of M to the refined bacterial ribosome particles improved the

resolution from approximately 15 Å to 7 Å at the 0.143 FSC criterion. This pipeline highlights RobPicker’s compatibility with modern refinement frameworks, enabling improved tilt-series alignment and enhanced structural resolution.

4.5.3 Data reweighting and label correction networks both contribute to RobPicker’s robustness

We conducted experiments on the SHREC 2019 Cryo-ET Challenge to validate the effectiveness of the data reweighting and label correction networks within the RobPicker framework. The benchmark consists of ten synthetic tomograms with ground-truth labels for particles across 12 distinct species, covering a wide range of sizes (Gubins et al., 2019). These particles were categorized by size into four groups: large, medium, small, and tiny (Fig. 4.4). The dataset was divided into training, validation, and test sets (Methods). Label maps for the validation and test sets were generated using the shape-based method. For the training set, label maps were generated under two settings: one using the shape-based method and the other using the sphere-based method, in order to evaluate RobPicker’s robustness to different levels of label noise.

We first compared RobPicker with DeepFinder (Moebel et al., 2021) on this benchmark. RobPicker consistently outperformed DeepFinder in terms of F1 score across all particle size categories—large, medium, small, and tiny—in the sphere-based experiments (Fig. 4.4a). Notably, RobPicker demonstrated substantial improvements over DeepFinder in detecting small and tiny particle species. For small species, such as 1u6g, 3d2f, 2cg9, and 3h84, RobPicker achieved F1 scores of 0.95, 0.90, 0.95, and 0.92, respectively, compared with DeepFinder’s scores of 0.55, 0.57, 0.38, and 0.34. For tiny species, including 3gl1, 3qm1, and 1s3x, RobPicker attained F1 scores of 0.49, 0.64, and 0.78, exceeding DeepFinder’s corresponding scores of 0.37, 0.30, and 0.01. Moreover, RobPicker also outperformed DeepFinder in the shape-based experiments (Fig. 4.4b), where the training labels were cleaner than those in the sphere-based experiments.

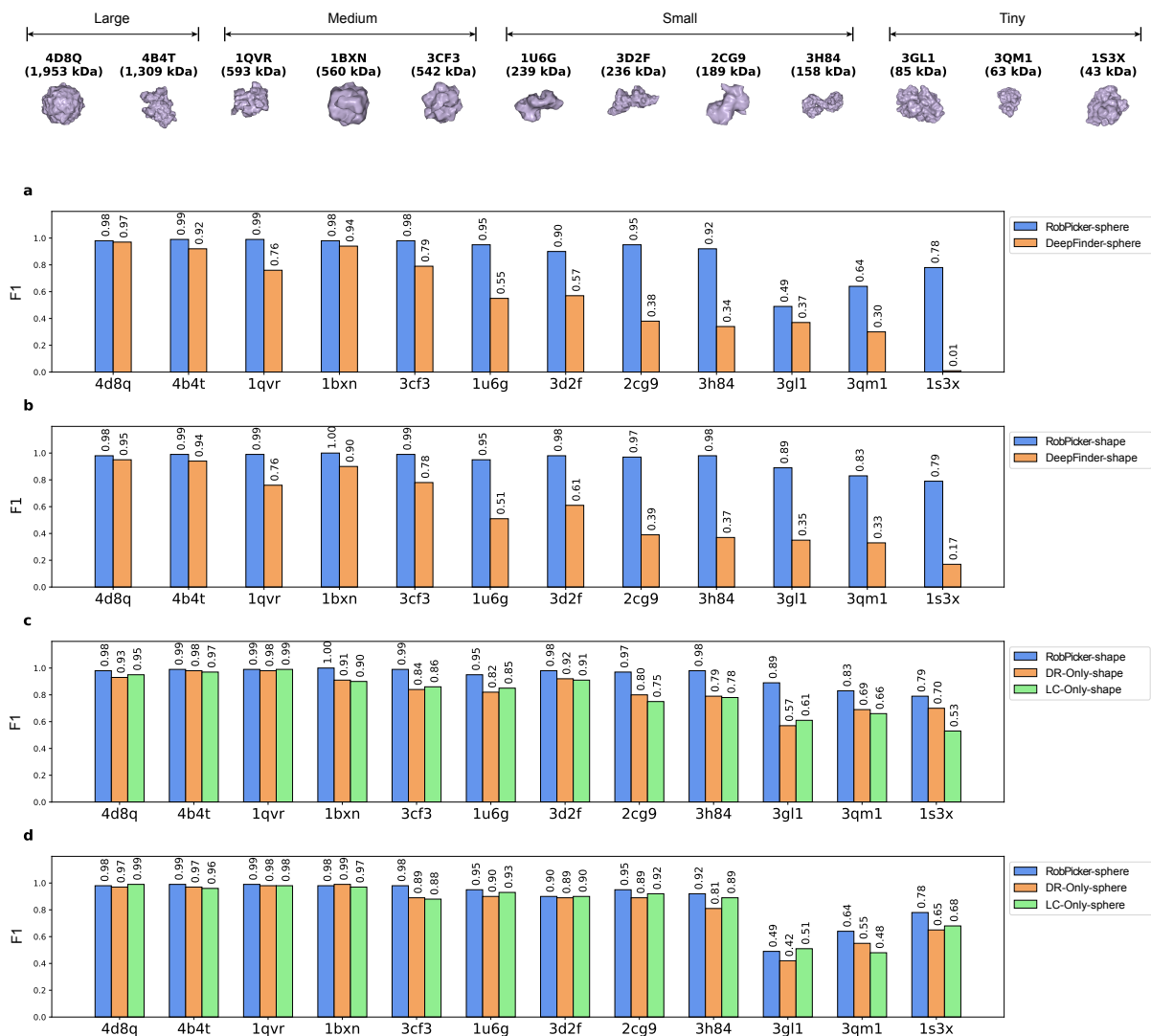


Figure 4.4: RobPicker significantly outperformed DeepFinder on the SHREC 2019 dataset. a-b, RobPicker outperformed DeepFinder across 12 distinct classes encompassing particles of different sizes. a, The training labels were sphere-based, where a sphere was used to approximate the occupancy of a particle. b, The training labels were shape-based, where particle contours were used to precisely annotate tomograms. The sphere-based labels were noisier than the shape-based labels but cheaper to generate. c-d, RobPicker integrated with both the data reweighting network (DRN) and label correction network (LCN) outperformed the ablations of RobPicker with only DRN or LCN, in both shape- and sphere-label experiments.

To evaluate the effectiveness of the data reweighting (DR) network and label correction (LC) network, we tested two variants of RobPicker: DR-Only, which performs data reweighting without label correction, and LC-Only, which applies label correction without data reweighting. RobPicker with both DR and LC networks significantly outperformed the two variants (Fig. 4.4c,d). For tiny species, RobPicker achieved F1 scores that were 56.1% and 45.9% higher (relative improvement) than DR-Only and LC-Only, respectively, on 3gl1; 20.3% and 25.8% higher on 3qm1; and 12.9% and 49.1% higher on 1s3x (Fig. 4.4c). Similarly, for small species, RobPicker outperformed DR-Only and LC-Only by 15.9% and 11.8%, respectively, on 1u6g; 6.5% and 7.7% on 3d2f; 21.3% and 29.3% on 2cg9; and 24.1% and 25.6% on 3h84. Improvements were also substantial for certain medium species, with RobPicker achieving 9.9% and 11.1% higher scores on 1bxn and 17.9% and 15.1% higher scores on 3cf3. These results highlight the importance of combining data reweighting and label correction to achieve robust performance in RobPicker.

The improvement of RobPicker over the two variants is more pronounced for smaller particles than for larger ones. Specifically, the average relative improvement of RobPicker over DR-Only for tiny, small, medium, and large species is 29.8%, 16.9%, 9.2%, and 3.2%, respectively, showing a clear decreasing trend. Similarly, the average relative improvement of RobPicker over LC-Only follows a similar pattern, with values of 40.2%, 18.6%, 8.7%, and 2.6% for tiny, small, medium, and large species, respectively, also reflecting a decline as particle size increases. These findings are consistent with those in the previous section and further underscore the effectiveness of RobPicker in detecting small and rare particles.

4.6 Model and Experimental Details

4.6.1 Model architecture

RobPicker uses a single 3D U-Net backbone whose encoder–decoder layers are shared between the segmentation and label-correction components. Each component then has its own task-specific prediction head. The segmentation component consists of the shared backbone

followed by a dedicated output head (including a softmax layer) that produces per-voxel class probabilities. The label-correction component reuses the same backbone but includes a separate, lightweight prediction head (comprising several convolutional layers and a softmax layer) that generates corrected label maps. In addition, the data-reweighting network is an independent, lightweight fully connected module that maps the per-patch loss to a scalar weight and does not share parameters with the 3D U-Net backbone.

We used PyTorch (Paszke et al., 2019) to implement the macromolecule segmentation network. For experiments comparing with DeepFinder, we used the 3D U-Net (Çiçek et al., 2016) following DeepFinder (Moebel et al., 2021). The input of the macromolecule segmentation network is a 3D tensor (e.g., $h \times w \times d$) representing a 3D tomogram. The output is a 4D tensor ($h \times w \times d \times c$) with c channels, where c equals the number of classes (number of species plus one background class). The model has four down-blocks, each containing four 3D convolutional layers (kernel size: $3 \times 3 \times 3$, stride: $1 \times 1 \times 1$, padding: $1 \times 1 \times 1$). Each convolutional layer is followed by a 3D batch normalization layer (Ioffe and Szegedy, 2015) and a ReLU activation function (Nair and Hinton, 2010). A 3D max-pooling layer (kernel size: 2, stride: 2, padding: 1, dilation: 1) is placed between the second and third convolutional layers in each down-block. The model predicts the probability of each voxel belonging to each class. Applying the arg max operation along the channel dimension yields the predicted class label for each voxel (see post-processing below). The down-blocks are followed by a bottleneck block with two 3D convolutional layers, and then four up-blocks. Each up-block contains a transposed 3D convolutional layer (kernel size: $2 \times 2 \times 2$, stride: $2 \times 2 \times 2$) and two 3D convolutional layers (kernel size: $3 \times 3 \times 3$, stride: $1 \times 1 \times 1$, padding: $1 \times 1 \times 1$). The final layer of this segmentation network is 3D convolutional layer (kernel size: $1 \times 1 \times 1$, stride: $1 \times 1 \times 1$). For experiment on D5 comparing RobPicker with DeePiCt, we used the same 3D U-Net as DeePiCt (de Teresa-Trueba et al., 2023), which has a similar architecture as DeepFinder and contains 41 million parameters.

The data reweighting network consists of a linear layer that expands the input dimension of 1 (i.e., the loss value) to a hidden dimension of 500, followed by a ReLU activation (Nair and

Hinton, 2010), and then a linear layer that reduces the hidden dimension of 500 to the output dimension of 1, followed by a sigmoid function that converts the output to a weight between 0 and 1. The data reweighting network contains only 1.5 thousand parameters. The label correction network takes as input the noisy label, denoted by $\mathbf{L}$ (a 4D tensor with c channels, where the channel dimension is one-hot), and the input tomogram (a 3D tensor). The label correction network begins with a sub-network that processes the input tomogram and outputs a feature map of the tomogram, denoted by $\mathbf{M}$. For efficiency considerations, this sub-network shares the same architecture and parameters as the macromolecule segmentation network. The segmentation network and the label correction network each have their own task-specific prediction heads. For the label correction component, the prediction head consists of a 3D convolutional layer that maps the input channel dimension of $\mathbf{M}$ to the number of output classes (kernel size 1, stride 1, padding 0), followed by a softmax operation to produce class probabilities. The result of the softmax is denoted by $\mathbf{S}$. Then, $\mathbf{S}$ is concatenated with the noisy label $\mathbf{L}$, resulting in a feature map with a channel dimension equal to twice the number of classes. A 3D convolution layer (kernel size 1, stride 1, padding 1) reduces the channel dimension to the number of classes, followed by a ReLU activation, another 3D convolution layer (kernel size 1, stride 1, padding 1), and a sigmoid operation. The output, denoted by α , serves as a mixing coefficient for combining the noisy label $\mathbf{L}$ and the softmax output $\mathbf{S}$ to compute the corrected label $\mathbf{C}$:

$$\mathbf{C} = \alpha \cdot \mathbf{S} + (1 - \alpha) \cdot \mathbf{L}. \quad (4.14)$$

Due to parameter sharing between the label correction network and the macromolecule segmentation network, the label correction network has only around 1,300 parameters. The small number of parameters in the label correction and data reweighting networks significantly reduces the computational complexity of the bi-level optimization in RobPicker.

4.6.2 Data preprocessing

Data structure. A tomogram is represented as a 3D tensor with shape $h \times w \times d$, while its label map is represented as a 4D tensor with shape $h \times w \times d \times c$, where c denotes the number of segmentation classes. For a label map denoted by M and a specific 3D coordinate (x, y, z) , let the ground-truth class be $e \in \{1, 2, \dots, c\}$. The label map is then constructed such that $M_{x,y,z,e} = 1$ and $M_{x,y,z,j} = 0$ for $j \neq e$.

Tomogram patching. During the training of deep learning models on large tomographic volumes (e.g., $500 \times 500 \times 500$), GPU memory capacity often becomes a bottleneck, frequently leading to out-of-memory errors. To address this limitation, we divided each tomogram into distinct 3D subtomograms (“patches”). Following DeepFinder (Moebel et al., 2021), we used a patch size of $64 \times 64 \times 64$. We also applied random rotations and random shifts for data augmentation during training. The corresponding labels were extracted using the same patch size. As a result, the segmentation network takes a 3D subtomogram as input and predicts labels for that subtomogram. The advantage of this strategy lies in its memory efficiency—only the current batch is loaded into GPU memory. This design ensures that the method remains adaptable across a range of GPU hardware. During inference, each test tomogram is also divided into subtomograms before being passed to the segmentation network. However, predictions near the boundaries of subtomograms may be less accurate due to insufficient tomographic context. To mitigate this issue, we extracted overlapping subtomograms (with an overlap of 20 voxels) and averaged the predictions in the overlapping regions of the segmentation maps.

Training label construction. The cryo-ET datasets D1-D5 provide particle locations as 3D coordinates. Following DeepFinder (Moebel et al., 2021), we converted these annotations into voxel-wise label maps using either (i) a shape-based approach, which places a species-specific mask derived from subtomogram averaging (Chen et al., 2019) at each particle location, or (ii) a faster sphere-based approach, which uses spherical masks but produces noisier labels. We split

each dataset into training, validation, and test sets following the standard protocol (Moebel et al., 2021). To evaluate RobPicker’s robustness to label noise while maintaining efficient preprocessing, we generated label maps for the (larger) training set using the sphere-based method and applied the more accurate shape-based method only to the (smaller) validation and test sets. This strategy minimizes subtomogram averaging, as it is performed on only a limited number of tomograms.

Resampling. To ensure that particle classes (i.e., macromolecule species) are balanced in the validation data, we used resampling when extracting subtomograms. Specifically, we first uniformly sampled a particle class from $\{1, 2, \dots, c\}$ and then sampled a particle from the selected class. A subtomogram centered on the sampled particle was then extracted for validation.

Data augmentation. During training, extensive data augmentation was applied to the 3D patches before they were fed into the segmentation model to improve generalization. Spatial augmentations were applied identically to both image and label volumes and included random spatial cropping, independent random flips along each axis, random 90° rotations in the x–y plane, and an additional small random rotation about the x axis. Reflection padding was used to prevent edge artifacts. In addition, Mixup (Zhang et al., 2018) was applied with a probability of 0.5: when enabled, pairs of samples within a batch were combined using a mixing coefficient drawn from a beta distribution, and both input volumes and corresponding labels were linearly interpolated. This augmentation strategy regularizes training by encouraging smoother decision boundaries.

Construction of a clean validation set. In practice, for many existing datasets where multiple tomograms have been annotated over time—often by different users or at different stages of a project—it is typically feasible to retrospectively inspect a small subset of these annotations and identify tomograms whose labels appear more reliable (e.g., fewer obvious false positives on membranes or ice, and more consistent particle placement). These tomograms can be designated

as the validation set with relatively modest effort: the user need not re-annotate the entire dataset, but only audit and rank a small number of existing annotations and select the cleanest ones.

We used this “selection from existing annotations” strategy to construct our validation sets.

Pixel size. A coarse pixel size is typically 10–20 Å/pixel, which is standard practice in cryo-ET analysis, whereas the raw pixel size typically ranges from 1 Å to 4 Å. In many deep learning–based pipelines, raw tomograms are binned by a factor of 3–6 to improve computational efficiency and enhance image contrast. Consequently, most machine learning methods for tomogram denoising and particle picking operate on tomograms binned to a coarse pixel size of approximately 10 Å to balance speed and performance, which is the setting used in most of our experiments. RobPicker is tolerant to a range of pixel sizes after training. For example, we trained RobPicker on tomograms with a coarse pixel size of 20 Å in D2 and subsequently performed inference using an even coarser pixel size (approximately 40 Å) to further accelerate prediction. Across these different pixel sizes, RobPicker consistently achieved strong particle-picking performance on multiple datasets and particle species, demonstrating its robustness to variations in pixel size.

4.6.3 Inference with a trained network

At inference, RobPicker converted dense network outputs into particle coordinates using one of two post-processing strategies, depending on the segmentation model. For the RobPicker model trained with particle masks, each normalized tomogram was partitioned into overlapping 3D subvolumes (128^3 voxels; overlap of 20 voxels) to mitigate boundary artifacts. The macro-molecule segmentation network was applied to each subvolume, and the per-patch predictions were reassembled into a full-volume probability map M by averaging class probabilities in overlapping regions. Voxel-wise classification was then performed by assigning each voxel to the most probable class, $C_{x,y,z} = \arg \max_i M_{x,y,z,i}$, yielding a discrete 3D label tensor C . Predicted voxels were subsequently grouped into individual particle instances using DBSCAN (Ester et al., 1996)

with a neighborhood radius of 1 voxel and a minimum cluster size of 5 voxels. Particle coordinates were computed as the centers of mass of the resulting clusters, with class labels inherited from the clustered voxels, consistent with the general strategy used in DeepFinder (Moebel et al., 2021).

For the 3D visualization of the particles, we used IMOD (Kremer et al., 1996) and ChimeraX (Meng et al., 2023).

4.6.4 Datasets

The D1 dataset was derived from a large-scale collection of annotated tomograms of the green alga *C. reinhardtii* (Kelley et al., 2026). Tilt series were acquired at a nominal pixel size of 1.96 Å, aligned, and reconstructed into tomograms at 4× binning, corresponding to a voxel size of 7.84 Å. We used the provided denoised tomograms with indices 137, 145, 178, 269, 295, 1287, 1317, 1333, 1694, 1959, 2025, 2038, 2048, 2158, 2224, 2252, 2814, and 2816. Tomograms 2814 and 2816 were designated as the test set, while the remaining 16 tomograms were used for training. For annotation, 80S ribosomes were identified using template matching with an 80S reference obtained by subtomogram averaging, which was subsequently applied across the entire dataset. ATP synthase particles were detected via template matching followed by 3D classification without alignment, producing multiple ATP synthase classes as well as “membrane mis-pick” classes. These classes were incorporated into a suppression and scoring scheme to reduce membrane-localized false positives.

D2 was generated to study co-translating 80S ribosomes in *S. cerevisiae* cells (Chang et al., 2025). It includes six tomograms from *S. cerevisiae* lamellae, in which PyTom (Bandyopadhyay et al., 2022; Harastani et al., 2025) template matching and manual expert annotation were used to annotate 80S ribosomes. Tomograms were collected at a voxel size of 2.63 Å and reconstructed at a voxel size of 10 Å in Warp (Tegunov and Cramer, 2019). All tomograms were denoised using IsoNet (Liu et al., 2022) to compensate for the missing-wedge effect.

For D3, we collected 28 tomograms from *E. coli* cells. These tomograms were collected at a voxel size of 1.66 Å, reconstructed and binned by 6 voxels to 9.98 Å using Warp, and were

not denoised further. The annotations for this dataset were generated using template matching and classification in RELION (Burt et al., 2024) to separate the 70S and 50S classes.

For D4, we collected five tomograms using the following procedure. *S. cerevisiae* cells in the logarithmic growth phase were vitrified and prepared for cryo-ET using cryo-FIB milling (Wagner et al., 2020). A ribosome particle picker was trained on this dataset and subsequently applied, without supervision, to eight additional tomograms of *S. cerevisiae* (also prepared using cryo-FIB), yielding ~23,000 ribosome coordinates. These particles were subjected to refinement in RELION 3 (Zivanov et al., 2018). The workflow included an initial refinement of all particles, followed by 3D classification. Particles belonging to the best-resolved class (~5,500 particles) were selected for subsequent refinement, resulting in a high-quality and representative *S. cerevisiae* ribosome map (resolution ~19 Å) shown in Fig. 4.3b.

The *M. pneumoniae* dataset used to demonstrate the pipeline in Fig. 4.3c was obtained from M (Tegunov et al., 2021). The same RobPicker model trained on D4 was used to pick ~5,300 particles from 14 tomograms of *M. pneumoniae*. These particles were subjected to initial refinement and classification in RELION 3 (Zivanov et al., 2018). From these, ~2,000 particles belonging to the highest-quality class were selected for multi-particle refinement in M (Tegunov et al., 2021). This refinement, coupled with tilt-series alignment fine-tuning, improved the ribosome map resolution from ~15 Å to 7 Å (Fig. 4.3c). All reported resolutions were estimated using the gold-standard FSC at the 0.143 criterion.

D5 was obtained from DeePiCt (de Teresa-Trueba et al., 2023). It includes 10 VPP tomograms of *S. pombe* cells. An iterative workflow was used to localize ribosomes and FAS in 4×-binned tomograms (13.48 Å voxel size). Manually curated template matching was performed for ribosomes, and non-exhaustive manual picking was conducted for FAS (step 1). The resulting annotations were used to train the 3D CNNs of DeePiCt (step 2). For ribosomes, step 2 was repeated three times (each time trained on the combined predictions of step 1 and the preceding round). Cumulative predictions were manually revised in tom_chooser in PyTom (for ribosomes) and in EMAN2 (Tang et al., 2007) (for FAS). This comprehensive pipeline ensured that the labels

were of high quality. We followed the original work (de Teresa-Trueba et al., 2023) and used five-fold cross-validation, splitting the tomograms into five equal-sized subsets. In each fold, one subset was reserved for testing, while the remaining four were used for training. For RobPicker, the training set (eight tomograms) was further divided into a new training set and a validation set in a 7:1 ratio, used for Stage I and Stage II optimization, respectively (Fig. 4.1a).

The SHREC 2019 challenge (Gubins et al., 2019) utilized 12 Protein Data Bank (PDB) identifiers (1bxn, 1qvr, 1s3x, 1u6g, 2cg9, 3cf3, 3d2f, 3gl1, 3h84, 3qm1, 4b4t, and 4d8q) to generate tomogram density maps. The density maps were subsequently transformed into tilt series of projection images, which were deliberately degraded with noise and contrast adjustments. The reconstructed tomograms from the tilt series have dimensions of $512 \times 512 \times 512$ with a voxel size of 1 nm. Each tomogram contains approximately 200 particles per species on average. Nine tomograms were provided for training, while the tenth tomogram was reserved for testing. For RobPicker, we partitioned the nine training tomograms into two subsets, assigning eight tomograms for training and one for validation. These subsets were used in Stage I and Stage II, respectively, of the training process shown in Fig. 4.1a.

4.6.5 Experimental settings

Loss function. We used the Dice loss function following DeepFinder (Moebel et al., 2021) and DeePiCt (de Teresa-Trueba et al., 2023). The Dice loss function is computed for each channel (i.e., class), and the final loss value is the average of the loss values across all channels. Specifically, for a ground-truth label map A (a 4D tensor of shape $h \times w \times d \times c$) and a predicted label map B , whose values lie in the interval $[0, 1]$, the Dice loss is computed as:

$$\text{Dice}(A, B) = \frac{1}{c} \sum_{j=1}^c \left(1 - \frac{2|A_j \cap B_j|}{|A_j| + |B_j|} \right) \quad (4.15)$$

where j denotes the index of the last dimension; the intersection is defined as $|A_j \cap B_j| = \sum_{x,y,z} A_{x,y,z,j} B_{x,y,z,j}$; and the union terms are defined as $|A_j| = \sum_{x,y,z} A_{x,y,z,j}$ and $|B_j| = \sum_{x,y,z} B_{x,y,z,j}$.

Evaluation metrics. We used the particle-picking F_β score to assess performance. The F_β is based on precision and recall:

$$F_\beta = (1 + \beta^2) \frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}} \quad (4.16)$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (4.17)$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4.18)$$

where TP, FP, FN denote the numbers of true positives, false positives, and false negatives, respectively. Predictions were classified as TP if their coordinates were within a distance threshold of a ground-truth annotation with the same class label. For D1-D5, we reported the F1 score to balance precision and recall. For D1, we used a distance threshold of 75 Å for 80S ribosomes and 40 Å for ATP synthase. For ribosomes in D2-D4, we used a distance threshold of 75 Å. For D5, we followed the evaluation metric of DeePiCt to use a distance threshold of 135 Å.

Hyperparameters and optimization. The stochastic gradient descent (SGD) algorithm (Amari, 1993; Ketkar and Ketkar, 2017) was used to update the model parameters. The optimizer used a learning rate of `lr=0.0001`, momentum set to `momentum=0.9`, and weight decay set to `weight_decay=0.0005`. Training was performed with a batch size of `batch_size=25`. The model was trained for 10,000 iterations (`train_iters=10000`), with a validation step performed every 500 iterations (`valid_iters=500`). The meta-learning components of our architecture used parameters such as `meta_alpha=0.0` and `lamda=1.0`, among other configurations, which are provided in the documentation of our code. For a dataset containing 10 tomograms, the training process typically takes a few hours on an Nvidia Tesla A100 (80 GB) GPU.

4.6.6 Baseline settings

3D U-Nets with class imbalance-aware loss functions. We used the same 3D U-Net architecture as RobPicker but replaced the loss function to address class imbalance using standard, manually specified reweighting strategies. We first employed a class-balanced weighted Dice loss, defined as

$$L_{\text{WDice}} = 1 - \frac{1}{N} \sum_c w_c \cdot \text{Dice}_c \quad (4.19)$$

where Dice_c denotes the Dice coefficient for class c , and w_c is a static class weight inversely proportional to the class frequency f_c (i.e., $w_c \propto 1/f_c$). The weights were normalized such that $\sum_c w_c = 1$. N denotes the batch size. This formulation increases the contribution of underrepresented particle classes during training.

Second, we used the standard focal loss

$$L_{\text{Focal}} = -\alpha(1 - p_t)^\gamma \log(p_t) \quad (4.20)$$

where p_t denotes the probability predicted by the model for the true class. We set the focusing parameter $\gamma = 2.0$ to down-weight well-classified examples and emphasize hard examples, and used a balance parameter $\alpha = 0.5$ to control the relative importance of positive and negative samples.

Third, we adopted Unified Focal Loss, which combines focal loss and Dice loss to jointly address class imbalance and segmentation overlap. The overall loss is defined as

$$L_{\text{Unified}} = L_{\text{WFocal}} + L_{\text{Dice}} \quad (4.21)$$

where the weighted focal term is given by

$$L_{\text{WFocal}} = -w_c(1 - p_t)^\gamma \log(p_t) \quad (4.22)$$

with a class-dependent weight

$$w_c = \frac{1}{f_c^\delta + \epsilon}. \quad (4.23)$$

Here, f_c denotes the class frequency, δ controls the degree of class reweighting, and ϵ is a small constant for numerical stability. We used $\delta = 0.6$, $\gamma = 2.0$, and set $\lambda = 1$ to balance the focal and Dice components.

Top-ranked Kaggle methods. We evaluated three top-ranked methods (Henkel and Khvedchenya, 2025; Luo and Wang, 2025; Ouertani et al., 2025) from a Kaggle competition for cryo-ET particle picking. For the D1–D5 comparisons, we followed the architectures and training configurations described in their public reports and released implementations (ran, 2025a,b,c).

Kaggle Rank-1 employs an ensemble comprising a ResNet-based 3D U-Net for semantic segmentation together with object detection models built on SegResNet (Myronenko, 2018) and DynUNet (Isensee et al., 2021) architectures. All models operate on volumetric input patches of size $96 \times 96 \times 96$. To address class imbalance, training uses a weighted cross-entropy loss that assigns a weight of 256 to positive voxels and a weight of 1 to background voxels. Optimization is performed using the Adam optimizer (Kingma and Ba, 2014) with a cosine learning rate schedule, a peak learning rate of 1×10^{-3} , and a batch size of 32.

Kaggle Rank-2 consists of an ensemble of lightweight volumetric segmentation models, including 3D U-Nets (Çiçek et al., 2016), VoxResNet (Chen et al., 2018a), VoxHRNet (Li et al., 2021), SegResNet (Myronenko, 2018), DenseVNet (Gibson et al., 2018), and UNet2E3D (Zhou et al., 2020). These models are trained on input patches of size $128 \times 256 \times 256$. The training objective combines Tversky loss (Salehi et al., 2017) and cross-entropy loss to balance overlap

accuracy and voxel-wise classification performance. Optimization is conducted using the AdamW optimizer (Loshchilov and Hutter, 2019b) with an initial learning rate of 1×10^{-3} and a batch size of 4.

Kaggle Rank-5 is based on a DeepFinder backbone (Moebel et al., 2021) with minor architectural modifications. Specifically, a three-dimensional batch normalization layer is applied at the input, and trilinear interpolation is used for both downsampling and upsampling operations, except for the final upsampling stage, which employs a transposed convolution. Models are trained on volumetric input patches of size $128 \times 128 \times 128$. The training objective is a label-smoothing cross-entropy loss with a smoothing factor of 0.01. Optimization is performed using the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 1×10^{-4} and a batch size of 4.

CryoSiam. CryoSiam (Stojanovska et al., 2025) is a self-supervised particle-picking framework pre-trained on simulated cryo-ET data and designed to generalize to experimental tomograms. The original CryoSiam model cannot be directly applied to particle picking on the D1–D5 datasets, as it does not include a segmentation head. To adapt CryoSiam to these datasets, we trained a lightweight segmentation head (approximately 0.1M parameters) on top of the CryoSiam-pretrained backbone (approximately 18M parameters) for each dataset. During this adaptation, the backbone was kept frozen and only the small segmentation head was updated. During both training and inference, volumetric patches of size 128^3 voxels were extracted with an overlap ratio of 0.5. For fine-tuning on D1–D5, we followed CryoSiam’s original loss formulation, which combines (i) a voxel-wise cross-entropy loss for classification accuracy, (ii) a generalized Dice loss to mitigate class imbalance, and (iii) a distance-to-background loss implemented as a mean squared error term to improve structural localization.

4.7 Discussion

Cellular cryo-tomograms contain rich structural information, yet many macromolecular species—especially small and sparse complexes—remain difficult to detect because they are

barely distinguishable from noise, imaging artifacts, and the crowded cellular background. This challenge is compounded by the fact that available annotations represent only a subset of the ground truth and inevitably contain both false positives and false negatives (Moebel et al., 2021). In this context, improving the detection of small and rare particles is particularly important for visual proteomics, where the goal is to study multiple interacting species rather than only the most abundant complexes. To address these limitations, we developed RobPicker, a meta-learning framework designed to be robust to two pervasive issues in cryo-ET particle picking: species imbalance and label noise. In our experiments on D1–D5, RobPicker consistently outperformed established supervised learning baselines such as DeepFinder and DeePiCt as well as the more recent Kaggle Rank-1, Rank-2, and Rank-5 pipelines originally developed for cryo-ET particle picking, with pronounced advantages for small and sparse particle classes.

We attribute these improvements to two tightly coupled components optimized through meta-learning: data reweighting and label correction. First, RobPicker learns instance-dependent weights that prioritize training patches most beneficial for improving validation performance, counteracting the tendency of standard supervised training, which often focuses on dominant, high signal-to-noise species (e.g., ribosomes) at the expense of smaller or rarer targets. Manual class weighting assigns higher weights to underrepresented classes based solely on class frequency, implicitly assuming that all samples within a class are equally informative. In cryo-ET, however, instances of the same particle class can vary substantially in quality due to differences in local signal-to-noise ratio, missing-wedge artifacts, and cellular context. Consequently, class-level reweighting often upweights poorly annotated or ambiguous samples alongside informative ones, limiting its effectiveness, particularly for small particles. By meta-optimizing instance-dependent weights based on performance on a cleaner validation subset, RobPicker instead learns to emphasize training samples that most improve generalization rather than uniformly reweighting an entire class.

Importantly, RobPicker also outperforms loss-based class-balancing strategies such as Focal Loss and Unified Focal Loss when using the same segmentation backbone. These methods

rely on error-driven “hard-example” reweighting, increasing the loss contribution of examples with high prediction error under the assumption that such examples are informative. In noisy cryo-ET tomograms, however, high-error examples frequently arise from mislabeled particles, ambiguous boundaries, missing-wedge artifacts, or background clutter rather than from genuinely informative training instances. As a result, Focal-style reweighting can inadvertently assign excessive weight to annotation errors, amplifying noise and leading to unstable or inconsistent performance across datasets. In contrast, RobPicker meta-learns example weights based on their utility for minimizing loss on a cleaner validation subset, allowing the model to prioritize training examples that demonstrably improve generalization. This validation-guided reweighting is tightly coupled with a label correction network that refines noisy annotations. As a result, RobPicker emphasizes informative minority examples while avoiding systematic reinforcement of noisy labels. Together, these mechanisms allow RobPicker to extract more reliable supervision from imperfect annotations and to handle class imbalance more robustly than static frequency-based weighting or error-driven reweighting strategies. This advantage is particularly pronounced for small and sparse macromolecular species, where supervision is both noisy and limited.

Second, RobPicker’s label correction mechanism enables effective learning from inexpensive but noisy supervision (e.g., sphere-based labels) by transforming it into a cleaner training signal before updating the segmentation network. This is particularly important in cryo-ET, where low signal-to-noise ratios, missing-wedge artifacts, and crowded cellular environments make annotations inherently imperfect and can cause naive supervised training to overfit to label artifacts. While self-supervised pretraining can, in principle, improve robustness to label noise by learning more general representations, in practice we found that RobPicker outperformed the self-supervised baseline CryoSiam on real cryo-ET datasets. These results highlight the difficulty of transferring self-supervised representations in cryo-ET: experimental tomograms vary widely in noise statistics, missing-wedge artifacts, acquisition conditions, and cellular context, and macromolecular appearance differs substantially across species. As a result, a self-supervised objective may preferentially encode dataset- or species-specific cues that do not reliably support

cross-domain particle discrimination. By explicitly correcting noisy supervision and optimizing with respect to a cleaner validation signal, RobPicker offers a more direct and data-grounded mechanism for mitigating label noise and improving generalization under challenging in situ conditions.

These design choices also help explain RobPicker’s advantage over the three top-ranked Kaggle pipelines on D1–D5. The Kaggle Rank-1, Rank-2, and Rank-5 methods (Henkel and Khvedchenya, 2025; Luo and Wang, 2025; Ouertani et al., 2025) use fixed training objectives and directly optimize against the provided annotations, without explicitly correcting label noise or using a cleaner validation subset to guide training. RobPicker instead jointly performs instance-dependent reweighting and label correction within a meta-learning framework that explicitly optimizes validation performance. This design improves robustness to annotation noise, tomogram-to-tomogram variability, and dataset-specific artifacts, while avoiding reliance on large model ensembles.

RobPicker does not eliminate the need for manual effort in training deep learning–based particle pickers; rather, it is designed to substantially reduce and better allocate that effort. With RobPicker, the user only needs to curate a comparatively small, higher-quality validation subset instead of exhaustively cleaning a large training set for supervised learning. The meta-learning framework of RobPicker then leverages this cleaner signal to guide label correction and data reweighting for the remaining, noisier annotations. In practice, such a validation set can be constructed with modest effort in two common scenarios. First, for projects that already contain annotations across multiple tomograms, a user can audit a subset and designate those with the most reliable labels (e.g., fewer membrane-associated false positives and more consistent particle placement) as validation data, which is often substantially less labor-intensive than re-annotating the full dataset. Second, a validation set can be constructed by carefully annotating a limited number of tomogram regions or subtomograms, for example through expert curation or double annotation with consensus. Such targeted annotation is increasingly feasible within modern cryo-ET processing pipelines. This design reflects the motivation behind meta-learning: it

is typically sample-efficient and can leverage a small amount of higher-quality supervision to improve training on much larger, noisier datasets.

Broader limitations and future directions for RobPicker, including robustness to domain shifts across laboratories and acquisition pipelines, preprocessing and normalization strategies, and integration with semi-supervised or self-supervised learning, are discussed in Chapter 5.

4.8 Data and Code Availability

Dataset D1 is publicly available in the EMPIAR database under accession number EMPIAR-11830 . The corresponding annotations are available at <https://github.com/Chromatin-Structure-Rhythms-Lab/ChlamyAnnotations>. D2 is available at EMPIAR-12534. D3 and D4 are available from the corresponding author upon reasonable request. D5 is available at EMPIAR-10988. The *M. pneumoniae* tomogram dataset is available at EMPIAR-10499. The dataset of the SHREC 2019 challenge can be accessed at the following URL: <http://www2.projects.science.uu.nl/shrec/cryo-et/2019/>.

4.8.1 Code Availability

The source code of RobPicker is available at <https://github.com/RobPicker/CryoET-RobPicker>

4.9 Toward General Adaptive Learning Systems

The three technical chapters adapt different components of the learning pipeline. EViT adapts the input sequence by reorganizing visual tokens. DrugChat adapts the prediction interface by conditioning on prompts and integrating molecular modalities. RobPicker adapts the supervision signal by correcting labels and reweighting examples. Together, these methods suggest a broader direction for deep learning: models should not treat data, tasks, or labels as fixed, but should learn how to use them adaptively.

Chapter 4, in full, has been submitted for publication of the material as it appears

in “RobPicker: A Meta Learning Framework for Robust Identification of Macromolecules in Cryo-Electron Tomograms” by Youwei Liang, Ramtin Hosseini, Hamidreza Rahmani, Digvijay Singh, Ruiyi Zhang, Li Zhang, Han Guo, Sang Choe, Joan Lee, Min Xu, Eran Segal, James Zou, James Williamson, Elizabeth Villa, Danielle Grotjahn, and Pengtao Xie. The dissertation author was the primary researcher and author of this paper.

Chapter 5

Conclusions and Future Directions

5.1 Conclusions

This dissertation first studied adaptive selection through EViT, an efficient token reorganization method for Vision Transformers. EViT addresses redundancy in visual recognition by observing that image patch tokens are not equally informative for a final prediction. Using class-token attention as an internal importance signal, EViT preserves the most attentive tokens and fuses less attentive tokens into compact contextual representations. This design reduces computation without introducing additional parameters or discarding all information from low-attention regions. Empirically, EViT improves the efficiency-accuracy tradeoff on ImageNet, including approximately 50% faster DeiT-S inference with only 0.3% top-1 accuracy reduction, and it can use the saved computation to process higher-resolution inputs. These results support the first conclusion of the dissertation: when input data contain redundancy, adaptive models can improve efficiency by deciding what information should receive detailed computation.

The dissertation next studied adaptive alignment through DrugChat, a multimodal large language model for molecular property prediction. DrugChat addresses heterogeneity on both sides of the prediction problem: molecules may be represented as graphs, images, strings, and natural-language descriptions, while downstream tasks may require binary decisions, structured labels, continuous decision scores, or free-form textual answers. DrugChat aligns graph-based and image-based molecular representations with an instruction-tuned language model by pro-

jecting molecular features into the language model’s token space and conditioning generation on task prompts. This unified molecule-prompt-answer formulation allows different molecular representations and output structures to be handled through a shared predictive interface. Across drug indication, pharmacodynamics, mechanism-of-action prediction, structured molecular benchmarks, and held-out-task FS-Mol evaluation, DrugChat shows that adaptive alignment can make pretrained models more flexible for scientific prediction tasks with heterogeneous inputs and outputs. These results support the second conclusion of the dissertation: heterogeneous representations, tasks, and output forms can be made more usable when they are aligned through a common representation and prediction interface.

Finally, this dissertation studied adaptive supervision through RobPicker, a robust framework for cryo-electron tomography particle picking under noisy and imbalanced annotations. Cryo-ET particle picking is limited by expensive labels, missing annotations, annotation noise, and large differences in particle frequency and size. RobPicker responds to this supervision heterogeneity by jointly training a segmentation network, a label correction network, and a data reweighting network through validation-guided bi-level optimization. The label correction network refines imperfect labels, while the data reweighting network learns how strongly each training subtomogram should influence the loss, including giving higher weight to rare or small particle classes. Across real cryo-ET datasets, RobPicker improves robustness over supervised, self-supervised, imbalance-aware, and top-ranked competition-derived ensemble baselines, with especially strong gains for difficult sparse classes. These results support the third conclusion of the dissertation: models should adapt not only their inputs and representations, but also the supervision signals that guide learning.

5.2 Limitations

The adaptive mechanisms studied in this dissertation improve efficiency, flexibility, and robustness, but they also introduce assumptions that should be made explicit. EViT is evaluated

mainly on image classification benchmarks, and its token-importance signal depends on class-token attention. This design is natural for standard Vision Transformers, but it may not transfer directly to dense prediction, detection, segmentation, video understanding, or architectures without a class token. In addition, EViT assumes that low-attention tokens can be compressed without losing task-critical evidence. This assumption is often reasonable for image classification, where background and repeated texture can be redundant, but it may be less reliable in settings where small visual details or spatially distributed signals are essential.

DrugChat inherits both the promise and the risks of large language models. Its free-form answers make molecular prediction more flexible, but the DrugBank evaluation still found incorrect-response rates of 37.5–41.2% across the four question types, and 2.04% of outputs were structurally unusable. These outputs therefore require verification by domain experts rather than being treated as ground truth. For structured tasks, the ratio of first-token “Yes” and “No” probabilities is a ranking-oriented decision score, not a calibrated probability of correctness. Conversational coherence has a similar boundary: DrugChat retained context in 95.9% of 148 follow-up turns, but mean follow-up factual correctness was 0.48. The FS-Mol evaluation holds out downstream tasks, and its stricter split also holds out FS-Mol training compounds, but approximately 95% of test targets remain represented in PubChem through other assays; it does not establish generalization to wholly novel biological targets. Moreover, the deterministic molecular image contains no chemical information beyond the 2D graph and no 3D geometry, so its benefit is attributable to a differently pretrained representation rather than a new spatial modality. Finally, the reasoning that connects molecular representations, prompts, and generated text remains largely opaque. These limitations are especially important in biomedical use, where plausible but incorrect explanations can mislead downstream analysis.

RobPicker reduces the burden of noisy and imbalanced cryo-ET supervision, but it does not remove the need for high-quality human input. Its meta-learning framework relies on a cleaner validation subset to guide label correction and data reweighting, and the quality, size, and representativeness of this subset can affect the learned supervision signal. RobPicker also

introduces additional training complexity and memory overhead through its auxiliary networks and bi-level optimization. More broadly, like most current particle pickers, it does not explicitly model the full variability of cryo-ET data across laboratories and acquisition pipelines. Differences in sample preparation, thickness, defocus, dose schemes, reconstruction procedures, missing-wedge artifacts, and structured cellular background can substantially change tomogram appearance and may limit out-of-the-box generalization to new settings.

5.3 Future Directions

Several future directions follow naturally from the dissertation theme. For adaptive selection, EViT-like token adaptation could be extended beyond image classification to dense prediction, video, and scientific imaging domains such as cryo-ET, microscopy, pathology, and molecular images. A particularly interesting direction is to adapt computation not only over two-dimensional image patches, but also over volumetric regions, temporal segments, or multiscale structures. Such systems could use task- or uncertainty-dependent signals to decide when to preserve fine details, when to compress redundant regions, and when to allocate more computation to rare but biologically important structures.

For adaptive alignment, DrugChat suggests a path toward more capable multimodal scientific foundation models. Future studies should evaluate genuinely novel protein targets and endpoint families, broaden training data to underexplored drug categories and less common molecular families, and test the model prospectively in expert-supervised workflows. Molecular predictions could also be grounded in complementary evidence from docking, molecular dynamics, quantum mechanical calculations, and structure-based simulation. Another important direction is expanding beyond small molecules to biologics, proteins, complexes, and formulation-dependent systems. Alongside this broader scope, future systems need calibrated uncertainty estimation, factuality verification, abstention and error-detection mechanisms, and explainability methods that help domain experts distinguish conversational fluency from reliable scientific

evidence.

For adaptive supervision, RobPicker points toward more general and label-efficient particle-picking systems. Robustness to domain shift could be improved through preprocessing and normalization strategies that harmonize intensity statistics, noise characteristics, missing-wedge effects, and acquisition-specific artifacts across tomograms. Training objectives could also explicitly encourage invariance to laboratory, sample-preparation, and reconstruction differences. Because RobPicker already uses a cleaner validation subset to guide learning, it is naturally compatible with semi-supervised and self-supervised strategies. Self-supervised pretraining on large collections of unlabeled tomograms, followed by meta-learned reweighting and label correction using a small curated validation subset, could reduce dependence on high-quality labels while improving transfer across datasets.

A broader goal is trustworthy adaptive scientific AI: systems that adaptively select, represent, and trust information while exposing uncertainty, supporting human validation, and remaining robust under heterogeneous real-world conditions. In such systems, adaptivity should not only improve benchmark performance, but also help scientists understand where a model is confident, where supervision is unreliable, and where human expertise should remain in the loop.

5.4 Closing Perspective

Across vision, molecular prediction, and cryo-electron tomography, this dissertation demonstrates that adaptive learning mechanisms can improve efficiency, flexibility, and robustness under heterogeneous data conditions. The results suggest that future deep learning systems should move beyond static processing pipelines and instead learn how to select, represent, and trust information in a data- and task-dependent manner.

Bibliography

- Cryo et 1st place solution. <https://www.kaggle.com/code/christofhenkel/cryo-et-1st-place-solution?scriptVersionId=223259615>, 2025a. Accessed: 2025-11-17.
- 7-model-6TTA-T4x2-patch-norm-v2. <https://www.kaggle.com/code/luoziqian/7-model-6tta-t4x2-patch-norm-v2/notebook?scriptVersionId=220452693>, 2025b. Accessed: 2025-11-17.
- inference script. <https://www.kaggle.com/code/youssefouertani/inference-script>, 2025c. Accessed: 2025-11-17.
- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. In *Advances in Neural Information Processing Systems*, 2019.
- Stephan Allenspach, Jan A Hiss, and Gisbert Schneider. Neural multi-task learning in drug design. *Nature Machine Intelligence*, 6(2):124–137, 2024.
- Shun-ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4-5):185–196, 1993.
- Hmrishav Bandyopadhyay, Zihao Deng, Leiting Ding, Sinuo Liu, Mostofa Rafid Uddin, Xiangrui Zeng, Sima Behpour, and Min Xu. Cryo-shift: reducing domain shift in cryo-electron subtomograms with unsupervised domain adaptation and randomization. *Bioinformatics*, 38(4):977–984, 2022.
- Priyanka Banerjee, Andreas O Eckert, Anna K Schrey, and Robert Preissner. Protox-ii: a web-server for the prediction of toxicity of chemicals. *Nucleic acids research*, 46(W1):W257–W263, 2018.

- Désirée Baumann and Knut Baumann. Reliable estimation of prediction errors for qsar models under model uncertainty using double cross-validation. *Journal of cheminformatics*, 6:1–19, 2014.
- Martin Beck and Wolfgang Baumeister. Cryo-electron tomography: can it reveal the molecular sociology of cells in atomic detail? *Trends in cell biology*, 26(11):825–837, 2016.
- Guy W. Bemis and Mark A. Murcko. The properties of known drugs. 1. molecular frameworks. *Journal of Medicinal Chemistry*, 39(15):2887–2893, 1996. doi: 10.1021/jm9602928. PMID: 8709122.
- Emmanuel Bengio, Pierre-Luc Bacon, Joelle Pineau, and Doina Precup. Conditional computation in neural networks for faster models. *arXiv preprint arXiv:1511.06297*, 2015.
- Tristan Bepler, Kotaro Kelley, Alex J Noble, and Bonnie Berger. Topaz-denoise: general deep denoising models for cryoem and cryoet. *Nature communications*, 11(1):5208, 2020.
- Casper Berger, Navya Premaraj, Raimond BG Ravelli, Kèvin Knoop, Carmen López-Iglesias, and Peter J Peters. Cryo-electron tomography on focused ion beam lamellae transforms structural cell biology. *Nature Methods*, 20(4):499–511, 2023.
- Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *International Conference on Machine Learning*, 2021.
- Jochen Böhm, Achilleas S Frangakis, Reiner Hegerl, Stephan Nickell, Dieter Typke, and Wolfgang Baumeister. Toward detecting and identifying macromolecules in a cellular context: template matching applied to electron tomograms. *Proceedings of the National Academy of Sciences*, 97(26):14245–14250, 2000.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf.
- Alister Burt, Bogdan Toader, Rangana Warshamanage, Andriko von Kügelgen, Euan Pyle, Jasenko Zivanov, Dari Kimanius, Tanmay AM Bharat, and Sjors HW Scheres. An image processing pipeline for electron cryo-tomography in relion-5. *FEBS Open Bio*, 14(11):1788–1804, 2024.

- Shujun Cai, Chen Chen, Zhi Yang Tan, Yinyi Huang, Jian Shi, and Lu Gan. Cryo-et reveals the macromolecular reorganization of *s. pombe* mitotic chromosomes in vivo. *Proceedings of the National Academy of Sciences*, 115(43):10977–10982, 2018.
- Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, volume 32, 2019.
- Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, 2020.
- Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.
- Ya-Ting Chang, Benjamin A Barad, Juliette Hamid, Hamidreza Rahmani, Brian M Zid, and Danielle A Grotjahn. Cytoplasmic ribosomes on mitochondria alter the local membrane environment for protein import. *Journal of Cell Biology*, 224(4):e202407110, 2025.
- Hao Chen, Qi Dou, Lequan Yu, Jing Qin, and Pheng-Ann Heng. Voxresnet: Deep voxelwise residual networks for volumetric brain segmentation. *NeuroImage*, 170:446–455, 2018a. doi: 10.1016/j.neuroimage.2017.04.041.
- Hongming Chen, Ola Engkvist, Yinhai Wang, Marcus Olivecrona, and Thomas Blaschke. The rise of deep learning in drug discovery. *Drug discovery today*, 23(6):1241–1250, 2018b.
- Huiyuan Chen, Feixiong Cheng, and Jing Li. idrug: Integration of drug repositioning and drug-target prediction via cross-network embedding. *PLoS computational biology*, 16(7): e1008040, 2020.
- Muyuan Chen, Wei Dai, Stella Y Sun, Darius Jonasch, Cynthia Y He, Michael F Schmid, Wah Chiu, and Steven J Ludtke. Convolutional neural networks for automated annotation of cellular cryo-electron tomograms. *Nature methods*, 14(10):983–985, 2017.
- Muyuan Chen, James M Bell, Xiaodong Shi, Stella Y Sun, Zhao Wang, and Steven J Ludtke. A complete data processing workflow for cryo-et and subtomogram averaging. *Nature methods*, 16(11):1161–1168, 2019.
- Tianqi Chen, Ian Goodfellow, and Jonathon Shlens. Net2net: Accelerating learning via knowledge transfer. *arXiv preprint arXiv:1511.05641*, 2015a.
- Wenlin Chen, James Wilson, Stephen Tyree, Kilian Weinberger, and Yixin Chen. Compressing neural networks with the hashing trick. In *International Conference on Machine Learning*,

2015b.

Zhengsu Chen, Lingxi Xie, Jianwei Niu, Xuefeng Liu, Longhui Wei, and Qi Tian. Visformer: The vision-friendly transformer. *arXiv preprint arXiv:2104.12533*, 2021.

Feixiong Cheng, Chuang Liu, Jing Jiang, Weiqiang Lu, Weihua Li, Guixia Liu, Weixing Zhou, Jin Huang, and Yun Tang. Prediction of drug-target interactions and drug repositioning via network-based inference. *PLoS computational biology*, 8(5):e1002503, 2012.

Feixiong Cheng, Rishi J Desai, Diane E Handy, Ruisheng Wang, Sebastian Schneeweiss, Albert-László Barabási, and Joseph Loscalzo. Network-based approach to prediction and population-based validation of in silico drug repurposing. *Nature communications*, 9(1):2691, 2018.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL <https://lmsys.org/blog/2023-03-30-vicuna/>.

Sang Keun Choe, Willie Neiswanger, Pengtao Xie, and Eric Xing. Betty: An automatic differentiation library for multilevel optimization. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=LV_MeMS38Q9.

Krzysztof Choromanski, Valerii Likhoshesterov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *Arxiv preprint 2104.13840*, 2021.

Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19*, pages 424–432. Springer, 2016.

Sergio Cruz-León, Tomáš Majtner, Patrick Hoffmann, Jan P Kreysing, Maarten W Tuijtel, Stefan L Schaefer, Katharina Geißler, Martin Beck, Beata Turoňová, and Gerhard Hummer. High-confidence 3d template matching for cryo-electron tomography. *Biophysical Journal*, 123(3): 183a, 2024.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9268–9277, 2019.

- George E Dahl, Navdeep Jaitly, and Ruslan Salakhutdinov. Multi-task neural networks for qsar predictions. *arXiv preprint arXiv:1406.1231*, 2014.
- Zihang Dai, Guokun Lai, Yiming Yang, and Quoc V Le. Funnel-transformer: Filtering out sequential redundancy for efficient language processing. *arXiv preprint arXiv:2006.03236*, 2020.
- Antoine Daina, Olivier Michielin, and Vincent Zoete. Swissadme: a free web tool to evaluate pharmacokinetics, drug-likeness and medicinal chemistry friendliness of small molecules. *Scientific reports*, 7(1):42717, 2017.
- Meindert Danhof, Elizabeth CM De Lange, Oscar E Della Pasqua, Bart A Ploeger, and Rob A Voskuyl. Mechanism-based pharmacokinetic-pharmacodynamic (pk-pd) modeling in translational drug research. *Trends in pharmacological sciences*, 29(4):186–191, 2008.
- Stéphane d’Ascoli, Hugo Touvron, Matthew Leavitt, Ari Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. *arXiv preprint arXiv:2103.10697*, 2021.
- Irene de Teresa-Trueba, Sara K Goetz, Alexander Mattausch, Frosina Stojanovska, Christian E Zimmerli, Mauricio Toro-Nahuelpan, Dorothy WC Cheng, Fergus Tollervey, Constantin Pape, Martin Beck, et al. Convolutional networks for supervised mining of molecular patterns within cellular context. *Nature Methods*, 20(2):284–294, 2023.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3): 297–302, 1945.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.
- Carl Edwards, ChengXiang Zhai, and Heng Ji. Text2mol: Cross-modal molecule retrieval with natural language queries. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 595–607, 2021.

- Fabian Eisenstein, Haruaki Yanagisawa, Hiroka Kashihara, Masahide Kikkawa, Sachiko Tsukita, and Radostin Danev. Parallel cryo electron tomography on in situ lamellae. *Nature Methods*, 20(1):131–138, 2023.
- Fabian Eisenstein, Yoshiyuki Fukuda, and Radostin Danev. Smart parallel automated cryo-electron tomography. *Nature Methods*, 21(9):1612–1615, 2024.
- Utz Ermel, Anchi Cheng, Jun Xi Ni, Jessica Gadling, Manasa Venkatakrishnan, Kira Evans, Jeremy Asuncion, Andrew Sweet, Janece Pourroy, Zun Shi Wang, Kandarp Khandwala, Benjamin Nelson, Dannielle McCarthy, Eric M. Wang, Richa Agarwal, and Bridget Carragher. A data portal for providing standardized annotations for cryo-electron tomography. *Nature Methods*, pages 1–3, 2024.
- Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pages 226–231, 1996.
- Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. *arXiv:2104.11227*, 2021.
- Shan Gao, Renmin Han, Xiangrui Zeng, Zhiyong Liu, Min Xu, and Fa Zhang. Macromolecules structural classification with a 3d dilated dense network in cryo-electron tomography. *IEEE/ACM transactions on computational biology and bioinformatics*, 19(1):209–219, 2021.
- Anna Gaulton, Louisa J Bellis, A Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, et al. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic acids research*, 40(D1): D1100–D1107, 2012.
- Chongjian Ge, Youwei Liang, Yibing Song, Jianbo Jiao, Jue Wang, and Ping Luo. Revitalizing CNN attention via transformers in self-supervised visual representation learning. In *Advances in Neural Information Processing Systems*, 2021.
- Eli Gibson, Wenqi Li, Carole H. Sudre, Lucas Fidon, Diya Shakir, Guotai Wang, Zach Eaton-Rosen, William Gray, Tom Doel, Yipeng Hu, Tim Whyntie, Parashkev Nachev, Marc Modat, Dean C. Barratt, Sebastien Ourselin, and M. Jorge Cardoso. Automatic multi-organ segmentation on abdominal ct with dense v-networks. *IEEE Transactions on Medical Imaging*, 37(8): 1822–1834, 2018. doi: 10.1109/TMI.2018.2806289.
- Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.
- Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer

- network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.
- Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*. MIT Press, 2016.
- Assaf Gottlieb, Gideon Y Stein, Eytan Ruppin, and Roded Sharan. Predict: a method for inferring novel drug indications with application to personalized medicine. *Molecular systems biology*, 7(1):496, 2011.
- Ben Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet’s clothing for faster inference. *arXiv preprint arXiv:2104.01136*, 2021.
- Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.
- Ilja Gubins, Gijs van der Schot, Remco C. Veltkamp, Friedrich Förster, Xuefeng Du, Xiangrui Zeng, Zhenxi Zhu, Lufan Chang, Min Xu, Emmanuel Moebel, Antonio Martinez-Sanchez, Charles Kervrann, Tuan M. Lai, Xusi Han, Genki Terashi, Daisuke Kihara, Benjamin A. Himes, Xiaohua Wan, Jingrong Zhang, Shan Gao, Yu Hao, Zhilong Lv, Xiaohua Wan, Zhidong Yang, Zijun Ding, Xuefeng Cui, and Fa Zhang. Classification in Cryo-Electron Tomograms. *Eurographics Workshop on 3D Object Retrieval*, 2019. ISSN 1997-0471. doi: 10.2312/3dor.20191061.
- Ilja Gubins, Marten L Chaillet, Gijs van Der Schot, Remco C Veltkamp, Friedrich Förster, Yu Hao, Xiaohua Wan, Xuefeng Cui, Fa Zhang, Emmanuel Moebel, et al. Shrec 2020: Classification in cryo-electron tomograms. *Computers & Graphics*, 91:279–289, 2020.
- Rohan Gupta, Devesh Srivastava, Mehar Sahu, Swati Tiwari, Rashmi K Ambasta, and Pravir Kumar. Artificial intelligence to deep learning: machine intelligence approach for drug discovery. *Molecular diversity*, 25:1315–1360, 2021.
- Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems*, volume 31, 2018.
- Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *arXiv preprint arXiv:2103.00112*, 2021a.
- Song Han, Jeff Pool, John Tran, and William J Dally. Learning both weights and connections for efficient neural networks. *arXiv preprint arXiv:1506.02626*, 2015.
- Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. Dynamic neural networks: A survey. *arXiv preprint arXiv:2102.04906*, 2021b.

- Mohamad Harastani, Gurudatt Patra, Charles Kervrann, and Mikhail Eltsov. Template learning: Deep learning with domain randomization for particle picking in cryo-electron tomography. *Nature Communications*, 16(1):8833, 2025.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollar, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.
- Esther Heid, Kevin P Greenman, Yunsie Chung, Shih-Cheng Li, David E Graff, Florence H Vermeire, Haoyang Wu, William H Green, and Charles J McGill. Chemprop: a machine learning package for chemical property prediction. *Journal of Chemical Information and Modeling*, 64(1):9–17, 2023.
- Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.
- Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *International conference on machine learning*, pages 2712–2721. PMLR, 2019.
- Christof Henkel and Eugene Khvedchenya. CZII - 1st Place Solution | Kaggle. <https://www.kaggle.com/competitions/czii-cryo-et-object-identification/writeups/daddies-1st-place-solution-segmentation-with-partl>, 2025. Accessed: 2025-11-17.
- Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. *arXiv preprint arXiv:2103.16302*, 2021.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5149–5169, 2021.
- Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *IEEE/CVF International Conference on Computer Vision*, 2019.
- Ting-Kuei Hu, Tianlong Chen, Haotao Wang, and Zhangyang Wang. Triple wins: Boosting accuracy, robustness and efficiency together by enabling input-adaptive inference. In *International*

Conference on Learning Representations, 2020a.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2020b. URL <https://openreview.net/forum?id=HJIWWJSFDH>.

Cheng-Yu Huang, Piotr Draczkowski, Yong-Sheng Wang, Chia-Yu Chang, Yu-Chun Chien, Yun-Han Cheng, Yi-Min Wu, Chun-Hsiung Wang, Yuan-Chih Chang, Yen-Chen Chang, et al. In situ structure and dynamics of an alphacoronavirus spike protein by cryo-et and cryo-em. *Nature communications*, 13(1):4877, 2022.

Qinwen Huang, Ye Zhou, and Alberto Bartesaghi. Milopyp: self-supervised molecular pattern mining and particle localization in situ. *Nature Methods*, 21(10):1863–1872, 2024.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

Fabian Isensee, Paul F. Jaeger, Simon A. A. Kohl, Jens Petersen, and Klaus H. Maier-Hein. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18(2):203–211, 2021. doi: 10.1038/s41592-020-01008-z.

Andrii Iudin, Paul K Korir, José Salavert-Torres, Gerard J Kleywegt, and Ardan Patwardhan. Empiar: a public archive for raw electron microscopy image data. *Nature methods*, 13(5): 387–388, 2016.

Stanisław Jastrzębski, Damian Leśniak, and Wojciech Marian Czarnecki. Learning to smile (s). *arXiv preprint arXiv:1602.06289*, 2016.

Zihang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Yujun Shi, Xiaojie Jin, Anran Wang, and Jiashi Feng. All tokens matter: Token labeling for training better vision transformers. *arXiv preprint arXiv:2104.10858*, 2021.

José Jiménez, Miha Skalic, Gerard Martinez-Rosell, and Gianni De Fabritiis. K deep: protein–ligand absolute binding affinity prediction via 3d-convolutional neural networks. *Journal of chemical information and modeling*, 58(2):287–296, 2018.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*, 2020.

Michael J Keiser, Vincent Setola, John J Irwin, Christian Laggner, Atheir I Abbas, Sandra J Hufeisen, Niels H Jensen, Michael B Kuijter, Roberto C Matos, Thuy B Tran, et al. Predicting new molecular targets for known drugs. *Nature*, 462(7270):175–181, 2009.

- Ron Kelley, Sagar Khavnekar, Ricardo D. Righetto, Jessica Heebner, Martin Obr, Xianjun Zhang, Saikat Chakraborty, Grigory Tagiltsev, Alicia K. Michael, Sofie van Dorst, Florent Waltz, Caitlyn L. McCafferty, Lorenz Lamm, Simon Zufferey, Philippe Van der Stappen, Hugo van den Hoek, Wojciech Wietrzynski, Pavol Harar, William Wan, John A.G. Briggs, Jürgen M. Plitzko, Benjamin D. Engel, and Abhay Kotecha. Toward community-driven visual proteomics with large-scale cryo-electron tomography of *Chlamydomonas reinhardtii*. *Molecular Cell*, 86(1):213–230.e7, 2026. ISSN 1097-2765. doi: 10.1016/j.molcel.2025.11.029. URL <https://doi.org/10.1016/j.molcel.2025.11.029>.
- Nikhil Ketkar and Nikhil Ketkar. Stochastic gradient descent. *Deep learning with Python: A hands-on introduction*, pages 113–132, 2017.
- Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, et al. Pubchem 2023 update. *Nucleic acids research*, 51(D1):D1373–D1380, 2023.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.
- Johannes Kirchmair, Andreas H Göller, Dieter Lang, Jens Kunze, Bernard Testa, Ian D Wilson, Robert C Glen, and Gisbert Schneider. Predicting drug metabolism: experiment and/or computation? *Nature reviews Drug discovery*, 14(6):387–404, 2015.
- Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.
- Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.
- Douglas B Kitchen, Hélène Decornez, John R Furr, and Jürgen Bajorath. Docking and scoring in virtual screening for drug discovery: methods and applications. *Nature reviews Drug discovery*, 3(11):935–949, 2004.
- James R Kremer, David N Mastronarde, and J Richard McIntosh. Computer visualization of three-dimensional image data using imod. *Journal of structural biology*, 116(1):71–76, 1996.
- Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
- Brent M Kuenzi, Jisoo Park, Samson H Fong, Kyle S Sanchez, John Lee, Jason F Kreisberg, Jianzhu Ma, and Trey Ideker. Predicting drug response and synergy using a deep learning model of human cancer cells. *Cancer cell*, 38(5):672–684, 2020.
- Lorenz Lamm, Ricardo D Righetto, Wojciech Wietrzynski, Matthias Pöge, Antonio Martinez-Sanchez, Tingying Peng, and Benjamin D Engel. Membrain: A deep learning-aided pipeline for detection of membrane proteins in cryo-electron tomograms. *Computer methods and programs in biomedicine*, 224:106990, 2022.
- Alon Lavie and Michael J Denkowski. The meteor metric for automatic evaluation of machine translation. *Machine translation*, 23:105–115, 2009.
- Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- Junnan Li, Richard Socher, and Steven C. H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*, 2020a.
- Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.
- Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.
- Yesu Li, Jonathan Cui, Yilun Sheng, Xiao Liang, Jingdong Wang, Eric I.-Chao Chang, and Yan Xu. Whole brain segmentation with full volume neural network. *Computerized Medical Imaging and Graphics*, 93:101991, 2021. ISSN 0895-6111. doi: <https://doi.org/10.1016/j.compmedimag.2021.101991>. URL <https://www.sciencedirect.com/science/article/pii/S0895611121001403>.
- Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. Learning from noisy labels with distillation. *Proceedings of the IEEE international conference on computer vision*, pages 1910–1918, 2017.
- Zeju Li, Konstantinos Kamnitsas, and Ben Glocker. Analyzing overfitting under class imbalance in neural networks for image segmentation. *IEEE transactions on medical imaging*, 40(3):1065–1077, 2020b.
- Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense

- object detection. In *IEEE ICCV*, 2017.
- Guole Liu, Tongxin Niu, Mengxuan Qiu, Yun Zhu, Fei Sun, and Ge Yang. Deepetpicker: Fast and accurate 3d particle picking for cryo-electron tomography using weakly supervised deep learning. *Nature Communications*, 15(1):2090, 2024a.
- Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=S1eYHoC5FX>.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024b.
- Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Haotang Deng, and Qi Ju. Fastbert: a self-distilling bert with adaptive inference time. *arXiv preprint arXiv:2004.02178*, 2020.
- Yun-Tao Liu, Heng Zhang, Hui Wang, Chang-Lu Tao, Guo-Qiang Bi, and Z Hong Zhou. Isotropic reconstruction for electron tomography with deep learning. *Nature communications*, 13(1): 6482, 2022.
- Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.
- Zhijian Liu. torchprofile. <https://github.com/zhijian-liu/torchprofile>, 2021.
- Zhiyuan Liu, An Zhang, Hao Fei, Enzhi Zhang, Xiang Wang, Kenji Kawaguchi, and Tat-Seng Chua. ProtT3: Protein-to-text generation for text-based protein understanding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5949–5966, Bangkok, Thailand, August 2024c. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.324. URL <https://aclanthology.org/2024.acl-long.324/>.
- Yu-Chen Lo, Stefano E Rensi, Wen Torng, and Russ B Altman. Machine learning in chemoinformatics and drug discovery. *Drug discovery today*, 23(8):1538–1546, 2018.
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019a.
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019b. URL <https://openreview.net/forum?id=Bkg6RiCqY7>.
- James Lu, Brendan Bender, Jin Y Jin, and Yuanfang Guan. Deep learning prediction of patient

- response time course from early data via neural-pharmacokinetic/pharmacodynamic modelling. *Nature machine intelligence*, 3(8):696–704, 2021.
- Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*, 2019.
- Vladan Lučić, Alexander Rigort, and Wolfgang Baumeister. Cryo-electron tomography: the challenge of doing structural biology in situ. *Journal of Cell Biology*, 202(3):407–419, 2013.
- Thomas Luechtefeld, Dan Marsh, Craig Rowlands, and Thomas Hartung. Machine learning of toxicological big data enables read-across structure activity relationships (rasar) outperforming animal test reproducibility. *Toxicological Sciences*, 165(1):198–212, 2018.
- Ping Luo, Zhenyao Zhu, Ziwei Liu, Xiaogang Wang, and Xiaoou Tang. Face model compression by distilling knowledge from neurons. In *AAAI Conference on Artificial Intelligence*, 2016.
- Ziqian Luo and Shuo Wang. CZII - 2nd Place Solution | Kaggle. <https://www.kaggle.com/competitions/czii-cryo-et-object-identification/writeups/luoziqian-lion-2nd-place-solution>, 2025. Accessed: 2025-11-17.
- Ziqian Luo, Xiangrui Zeng, Zhipeng Bao, and Min Xu. Deep learning-based strategy for macromolecules classification with imbalanced data from cellular electron cryotomography. In *2019 International joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2019.
- Judith C Madden, Steven J Enoch, Alicia Paini, and Mark TD Cronin. A review of in silico tools as alternatives to animal testing: principles, resources and applications. *Alternatives to Laboratory Animals*, 48(4):146–172, 2020.
- Julia Mahamid, Stefan Pfeffer, Miroslava Schaffer, Elizabeth Villa, Radostin Danev, Luis Kuhn Cuellar, Friedrich Förster, Anthony A Hyman, Jürgen M Plitzko, and Wolfgang Baumeister. Visualizing the molecular sociology at the hela cell nuclear periphery. *Science*, 351(6276):969–972, 2016.
- Andreas Mayr, Günter Klambauer, Thomas Unterthiner, and Sepp Hochreiter. Deeptox: toxicity prediction using deep learning. *Frontiers in Environmental Science*, 3:80, 2016.
- Andreas Mayr, Günter Klambauer, Thomas Unterthiner, Marvin Steijaert, Jörg K Wegner, Hugo Ceulemans, Djork-Arné Clevert, and Sepp Hochreiter. Large-scale comparison of machine learning methods for drug target prediction on chembl. *Chemical science*, 9(24):5441–5451, 2018.
- Caitlyn L McCafferty, Sven Klumpe, Rommie E Amaro, Wanda Kukulski, Lucy Collinson, and Benjamin D Engel. Integrating cellular electron microscopy with multimodal data to explore

- biology across space and time. *Cell*, 187(3):563–584, 2024.
- David Mendez, Anna Gaulton, A Patrícia Bento, Jon Chambers, Marleen De Veij, Eloy Félix, María Paula Magariños, Juan F Mosquera, Prudence Mutowo, Michał Nowotka, et al. ChEMBL: towards direct deposition of bioassay data. *Nucleic acids research*, 47(D1):D930–D940, 2019.
- Elaine C Meng, Thomas D Goddard, Eric F Pettersen, Greg S Couch, Zach J Pearson, John H Morris, and Thomas E Ferrin. Ucsf chimeraX: Tools for structure building and analysis. *Protein Science*, 32(11):e4792, 2023.
- Emmanuel Moebel and Charles Kervrann. Towards unsupervised classification of macromolecular complexes in cryo electron tomography: Challenges and opportunities. *Computer Methods and Programs in Biomedicine*, 225:107017, 2022.
- Emmanuel Moebel, Antonio Martinez-Sanchez, Lorenz Lamm, Ricardo D Righetto, Wojciech Wietrzynski, Sahradha Albert, Damien Larivière, Eric Fourmentin, Stefan Pfeffer, Julio Ortiz, et al. Deep learning improves macromolecule identification in 3d cellular cryo-electron tomograms. *Nature methods*, 18(11):1386–1394, 2021.
- Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440*, 2016.
- Andriy Myronenko. 3d mri brain tumor segmentation using autoencoder regularization. In *BrainLes@MICCAI*, 2018. URL <https://api.semanticscholar.org/CorpusID:53104235>.
- Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.
- Eva Nogales and Julia Mahamid. Bridging structural and cell biology with cryo-electron microscopy. *Nature*, 628(8006):47–56, 2024.
- Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021.
- Ryosaku Ota and Fumiyoshi Yamashita. Application of machine learning techniques to the analysis and prediction of drug pharmacokinetics. *Journal of Controlled Release*, 352:961–969, 2022.
- Youssef Ouertani, Ezzeddine Ouertani, Wertani Mehdi, and Aymen Souid. CZII - 5th Place Solution | Kaggle. <https://www.kaggle.com/competitions/czii-cryo-et-object-identification/writeups/youssef-ouertani-5th-place-solution>, 2025. Accessed: 2025-11-17.
- Hakime Öztürk, Arzucan Özgür, and Elif Ozkirimli. Deepdta: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, 2018.

- Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. Videomoco: Contrastive video representation learning with temporally adversarial examples. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.
- Douglas EV Pires, Tom L Blundell, and David B Ascher. pkcsm: predicting small-molecule pharmacokinetic and toxicity properties using graph-based signatures. *Journal of medicinal chemistry*, 58(9):4066–4072, 2015.
- Euan Pyle and Giulia Zanetti. Current data processing strategies for cryo-electron tomography and subtomogram averaging. *Biochemical journal*, 478(10):1827–1845, 2021.
- Euan Pyle, Elizabeth A Miller, and Giulia Zanetti. Cryo-electron tomography reveals how copii assembles on cargo-containing membranes. *Nature Structural & Molecular Biology*, 32(3): 513–519, 2025.
- Yujie Qian, Zhening Li, Zhengkai Tu, Connor Coley, and Regina Barzilay. Predictive chemistry augmented with text retrieval. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12731–12745, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.784. URL <https://aclanthology.org/2023.emnlp-main.784/>.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.
- Bharath Ramsundar, Steven Kearnes, Patrick Riley, Dale Webster, David Konerding, and Vijay Pande. Massively multitask networks for drug discovery. *arXiv preprint arXiv:1502.02072*, 2015.
- Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *arXiv preprint arXiv:2106.02034*, 2021.

- Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL <https://arxiv.org/abs/1908.10084>.
- Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pages 4334–4343. PMLR, 2018.
- Gavin Rice, Thorsten Wagner, Markus Stabrin, Oleg Sitsel, Daniel Prumbaum, and Stefan Raunser. Tomotwin: generalized 3d localization of macromolecules in cryo-electron tomograms with structural data mining. *Nature methods*, 20(6):871–880, 2023.
- Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.
- Michael S Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: Adaptive space-time tokenization for videos. In *Advances in Neural Information Processing Systems*, 2021.
- Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. Tversky loss function for image segmentation using 3d fully convolutional deep networks. *International workshop on machine learning in medical imaging*, pages 379–387, 2017.
- Philipp Seidl, Andreu Vall, Sepp Hochreiter, and Günter Klambauer. Enhancing activity prediction models in drug discovery with the ability to understand human language. In *International Conference on Machine Learning*, pages 30458–30490. PMLR, 2023.
- Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems*, 32, 2019.
- Rodolfo S Simões, Vinicius G Maltarollo, Patricia R Oliveira, and Kathia M Honorio. Transfer and multi-task learning in qsar modeling: advances and challenges. *Frontiers in pharmacology*, 9:74, 2018.
- Suraj Srinivas and R Venkatesh Babu. Data-free parameter pruning for deep neural networks. *arXiv preprint arXiv:1507.06149*, 2015.
- Megan Stanley, John F Bronskill, Krzysztof Maziarz, Hubert Misztela, Jessica Lanini, Marwin Segler, Nadine Schneider, and Marc Brockschmidt. Fs-mol: A few-shot learning dataset of molecules. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

- Marta M Stepniewska-Dziubinska, Piotr Zielenkiewicz, and Pawel Siedlecki. Development and evaluation of a deep learning model for protein–ligand binding affinity prediction. *Bioinformatics*, 34(21):3666–3674, 2018.
- Teague Sterling and John J Irwin. Zinc 15–ligand discovery for everyone. *Journal of chemical information and modeling*, 55(11):2324–2337, 2015.
- Frosina Stojanovska, Ricardo M Sanchez, Rasmus K Jensen, Julia Mahamid, Anna Kreshuk, and Judith B Zaugg. CryoSiam: self-supervised representation learning for automated analysis of cryo-electron tomograms. *bioRxiv*, pages 2025–11, 2025.
- Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *IEEE/CVF International Conference on Computer Vision*, 2017.
- Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *IEEE/CVF International Conference on Computer Vision*, 2019.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS’14, page 3104–3112, Cambridge, MA, USA, 2014. MIT Press.
- Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, 2019.
- Guang Tang, Liwei Peng, Philip R Baldwin, Deepinder S Mann, Wen Jiang, Ian Rees, and Steven J Ludtke. Eman2: an extensible image processing suite for electron microscopy. *Journal of structural biology*, 157(1):38–46, 2007.
- Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*, 2022.
- Dimitry Tegunov and Patrick Cramer. Real-time cryo-electron microscopy data preprocessing with warp. *Nature methods*, 16(11):1146–1152, 2019.
- Dimitry Tegunov, Liang Xue, Christian Dienemann, Patrick Cramer, and Julia Mahamid. Multi-particle cryo-em refinement with m visualizes ribosome-antibiotic complex at 3.5 Å in cells. *Nature methods*, 18(2):186–193, 2021.
- Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. Large language models in medicine. *Nature medicine*, 29(8):1930–1940, 2023.

- Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, 2021a.
- Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. *arXiv preprint arXiv:2103.17239*, 2021b.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- Alexander Tropsha. Best practices for qsar model development, validation, and exploitation. *Molecular informatics*, 29(6-7):476–488, 2010.
- Alexander Tropsha, Olexandr Isayev, Alexandre Varnek, Gisbert Schneider, and Artem Cherkasov. Integrating qsar modelling and deep learning in drug discovery: the emergence of deep qsar. *Nature Reviews Drug Discovery*, 23(2):141–155, 2024.
- Martin Turk and Wolfgang Baumeister. The promise and the challenges of cryo-electron tomography. *FEBS letters*, 594(20):3243–3261, 2020.
- Thomas Unterthiner, Andreas Mayr, Günter Klambauer, Marvin Steijaert, Jörg K Wegner, Hugo Ceulemans, and Sepp Hochreiter. Deep learning as an opportunity in virtual screening. In *Proceedings of the deep learning workshop at NIPS*, volume 27, pages 1–9. MIT Press Cambridge, MA, United States, 2014.
- Jessica Vamathevan, Dominic Clark, Paul Czodrowski, Ian Dunham, Edgardo Ferran, George Lee, Bin Li, Anant Madabhushi, Parantu Shah, Michaela Spitzer, et al. Applications of machine learning in drug discovery and development. *Nature reviews Drug discovery*, 18(6):463–477, 2019.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017a.
- Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017b.
- Felix R Wagner, Reika Watanabe, Ruud Schampers, Digvijay Singh, Hans Persoon, Miroslava Schaffer, Peter Fruhstorfer, Jürgen Plitzko, and Elizabeth Villa. Preparing samples from whole cells using focused-ion-beam milling for cryo-electron tomography. *Nature protocols*, 15(6): 2041–2070, 2020.

- Thorsten Wagner and Stefan Raunser. The evolution of sphere-cryolo particle picking and its application in automated cryo-em processing workflows. *Communications biology*, 3(1):61, 2020.
- Sinong Wang, Belinda Z Li, Madian Khabisa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020a.
- Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *IEEE/CVF International Conference on Computer Vision*, 2021a.
- Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. In *European Conference on Computer Vision*, 2018.
- Yue Wang, Jianghao Shen, Ting-Kuei Hu, Pengfei Xu, Tan Nguyen, Richard Baraniuk, Zhangyang Wang, and Yingyan Lin. Dual dynamic inference: Enabling more efficient, adaptive, and controllable deep inference. *IEEE Journal of Selected Topics in Signal Processing*, 2020b.
- Yulin Wang, Rui Huang, Shiji Song, Zeyi Huang, and Gao Huang. Not all images are worth 16x16 words: Dynamic vision transformers with adaptive sequence length. *arXiv preprint arXiv:2105.15075*, 2021b.
- Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021c.
- Yuyang Wang, Jianren Wang, Zhonglin Cao, and Amir Barati Farimani. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, 4(3): 279–287, 2022a.
- Ziyu Wang, Wenhao Jiang, Yiming Zhu, Li Yuan, Yibing Song, and Wei Liu. Dynamixer: A vision mlp architecture with dynamic mixing. *arXiv preprint arXiv:2201.12083*, 2022b.
- Abigail JI Watson and Alberto Bartesaghi. Advances in cryo-et data processing: meeting the demands of visual proteomics. *Current Opinion in Structural Biology*, 87:102861, 2024.
- Simon Wiedemann and Reinhard Heckel. A deep learning method for simultaneous denoising and missing wedge reconstruction in cryogenic electron tomography. *Nature Communications*, 15(1):8255, 2024.
- Oliver Wieder, Stefan Kohlbacher, Mélaïne Kuenemann, Arthur Garon, Pierre Ducrot, Thomas Seidel, and Thierry Langer. A compact review of molecular property prediction with graph neural networks. *Drug Discovery Today: Technologies*, 37:1–12, 2020.

- David S Wishart, Craig Knox, An Chi Guo, Savita Shrivastava, Murtaza Hassanali, Paul Stothard, Zhan Chang, and Jennifer Woolsey. Drugbank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic acids research*, 34(suppl_1):D668–D672, 2006.
- Felix Wong, Erica J Zheng, Jacqueline A Valeri, Nina M Donghia, Melis N Anahtar, Satotaka Omori, Alicia Li, Andres Cubillos-Ruiz, Aarti Krishnan, Wengong Jin, et al. Discovery of a structural class of antibiotics with explainable deep learning. *Nature*, 626(7997):177–185, 2024.
- Yichen Wu, Jun Shu, Qi Xie, Qian Zhao, and Deyu Meng. Learning to purify noisy labels via meta soft label corrector. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):10388–10396, 2021.
- Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.
- Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.
- Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Advances in Neural Information Processing Systems*, 2021.
- Guoli Xiong, Zhenxing Wu, Jiakai Yi, Li Fu, Zhijiang Yang, Changyu Hsieh, Mingzhu Yin, Xiangxiang Zeng, Chengkun Wu, Aiping Lu, et al. Admetlab 2.0: an integrated online platform for accurate and comprehensive predictions of admet properties. *Nucleic acids research*, 49(W1):W5–W14, 2021.
- Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.
- Minghao Xu, Xinyu Yuan, Santiago Miret, and Jian Tang. Protst: Multi-modality learning of protein sequences and biomedical texts. In *International Conference on Machine Learning*, pages 38749–38767. PMLR, 2023.
- Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. *arXiv preprint arXiv:2104.06399*, 2021.
- Riqiang Yan and Robert Vassar. Targeting the β secretase bace1 for alzheimer’s disease therapy. *The Lancet Neurology*, 13(3):319–329, 2014.
- Rui Yan, Singanallur V Venkatakrishnan, Jun Liu, Charles A Bouman, and Wen Jiang. Mbir:

- A cryo-et 3d reconstruction method that effectively minimizes missing wedge artifacts and restores missing information. *Journal of structural biology*, 206(2):183–192, 2019.
- Jason H Yang, Sarah N Wright, Meagan Hamblin, Douglas McCloskey, Miguel A Alcantar, Lars Schrübbers, Allison J Lopatkin, Sangeeta Satish, Amir Nili, Bernhard O Palsson, et al. A white-box machine learning approach for revealing antibiotic mechanisms of action. *Cell*, 177(6):1649–1661, 2019a.
- Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388, 2019b.
- Michael Yeung, Evis Sala, Carola-Bibiane Schönlieb, and Leonardo Rundo. Unified focal loss: Generalising dice and cross entropy-based losses to handle class imbalanced medical image segmentation. *Computerized Medical Imaging and Graphics*, 95:102026, 2022.
- Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021.
- Xiangrui Zeng, Anson Kahng, Liang Xue, Julia Mahamid, Yi-Wei Chang, and Min Xu. High-throughput cryo-et structural pattern mining by unsupervised deep iterative subtomogram clustering. *Proceedings of the National Academy of Sciences*, 120(15):e2213149120, 2023.
- Xiangxiang Zeng, Siyi Zhu, Xiangrong Liu, Yadi Zhou, Ruth Nussinov, and Feixiong Cheng. deepdr: a network-based deep learning approach to in silico drug repositioning. *Bioinformatics*, 35(24):5191–5198, 2019.
- Xiangxiang Zeng, Hongxin Xiang, Linhui Yu, Jianmin Wang, Kenli Li, Ruth Nussinov, and Feixiong Cheng. Accurate prediction of molecular properties and drug targets using a self-supervised image representation learning framework. *Nature Machine Intelligence*, 4(11):1004–1016, 2022a.
- Yuchen Zeng, Gregory Howe, Kai Yi, Xiangrui Zeng, Jing Zhang, Yi-Wei Chang, and Min Xu. Unsupervised domain alignment based open set structural recognition of macromolecules captured by cryo-electron tomography. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 106–110. IEEE, 2021.
- Zheni Zeng, Yuan Yao, Zhiyuan Liu, and Maosong Sun. A deep-learning system bridging molecule structure and biomedical text with comprehension comparable to human professionals. *Nature communications*, 13(1):862, 2022b.
- Biao Zhang, Zhongtao Liu, Colin Cherry, and Orhan Firat. When scaling meets llm finetuning:

- The effect of data, model and finetuning method. In *International Conference on Learning Representations*, 2024.
- Haonan Zhang, Yan Li, Yanan Liu, Dongyu Li, Lin Wang, Kai Song, Keyan Bao, and Ping Zhu. A method for restoring signals and revealing individual macromolecule states in cryo-et, rest. *Nature Communications*, 14(1):2937, 2023.
- Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. URL <https://openreview.net/forum?id=r1Ddp1-Rb>.
- Xiao-Chen Zhang, Cheng-Kun Wu, Zhi-Jiang Yang, Zhen-Xing Wu, Jia-Cai Yi, Chang-Yu Hsieh, Ting-Jun Hou, and Dong-Sheng Cao. Mg-bert: leveraging unsupervised atomic representation learning for molecular property prediction. *Briefings in bioinformatics*, 22(6):bbab152, 2021.
- Rongjian Zhao, Buyue Qian, Xianli Zhang, Yang Li, Rong Wei, Yang Liu, and Yinggang Pan. Rethinking dice loss for medical image segmentation. In *2020 IEEE international conference on data mining (ICDM)*, pages 851–860. IEEE, 2020.
- Guoqing Zheng, Ahmed Hassan Awadallah, and Susan Dumais. Meta label correction for noisy label learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):11053–11061, 2021.
- Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021.
- Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. *IEEE Transactions on Medical Imaging*, 39(6):1856–1867, 2020. doi: 10.1109/TMI.2019.2959609.
- Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *International Conference on Learning Representations*, 2024.
- Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.
- Jasenko Zivanov, Takanori Nakane, Björn O Forsberg, Dari Kimanius, Wim JH Hagen, Erik Lindahl, and Sjors HW Scheres. New tools for automated high-resolution cryo-em structure determination in relion-3. *elife*, 7:e42166, 2018.
- Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: Learning dense volumetric segmentation from sparse annotation. In *International*

Conference on Medical Image Computing and Computer-Assisted Intervention, 2016. URL <https://api.semanticscholar.org/CorpusID:2164893>.