University of California
Santa Barbara

Understanding and Facilitating Human-AI Teaming for Real-World Computer Vision Tasks

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

Doctor of Philosophy
in
Media Arts and Technology

by

Chengyuan Xu 许程远

Committee in charge:

Professor Tobias Höllerer, Chair
Professor Jennifer Jacobs
Dr. Curtis McCully
Professor Marko Peljhan

June 2024

The Dissertation of Chengyuan Xu 许程远 is approved.

Professor Jennifer Jacobs

Dr. Curtis McCully

Professor Marko Peljhan

Professor Tobias Höllerer, Committee Chair

June 2024

Understanding and Facilitating Human-AI Teaming for Real-World Computer Vision Tasks

by

Chengyuan Xu 许程远

Dedicated To

My parents, Yongmei Cheng and Zhilin Xu 程咏梅 许治林,

you shaped me.

My wife, Huan Liu 刘欢, you bettered me.

My daughter, Hedy Xu 许合意, you completed me.

Acknowledgements

Working towards my Ph.D. has been a fantastic journey thanks to the great company I've had along the way. I am grateful to each person who has been a part of this journey.

I extend my heartfelt gratitude to my advisor, Prof. Tobias Höllerer. Your mentorship and support have not only shaped me into a better researcher but also into a better person. Your kindness encourages everyone around you each day. To my dissertation committee members, Prof. Jennifer Jacobs, Dr. Curtis McCully, and Prof. Marko Peljhan, I am deeply grateful for your selfless guidance and moral support over the years. To the members of the Four Eyes Lab and the Expressive Computation Lab, you are my family abroad. Keep up the good work and nurture our collaborative community that never fails to inspire and support each other. I am also thankful for the insights I received from Professors Pradeep Sen, Marcos Novak, Matthew Turk, Karl Yerkes, Misha Sra, Yon Visell, Michael Stohl, and Paul Amar during my Ph.D. studies.

To my dear friends, Yuxiang, Zi Wang, Ekta, Abhishek, Chris, Steve, Yitian, Zhenyu, Yan, Chong, Da Zhang, Wei-Ting, Kuo-Kai, Ken, Weiyi, Jiajia, Yang Qiu, Xiuhe, Jiaxiang, Wenhu, Myungin, Mengyu, Weihao, Qiaodong, Han-Wei, Ehsan, Mengjia, Weidi, Jungah, Jing, Mert, Jieliang, Yi Ding, Peter, You-Jin, Alex, Noah, Radha, Sam, Ana, Ashley, Kangyou, Joyce, Zheng, Songyin, Yifan, Shanxiu, Yaoyi, Boning, Andy, Daichi, Kuo-Chin, Melissa, Lizhong, Katie, and many others, thank you for being the highlights of this seven-year journey. The laughs we shared and the challenges we overcame together have made Santa Barbara an unforgettable life adventure.

Curriculum Vitæ
Chengyuan Xu 许程远

Education

2017 - 2024	Ph.D. in Media Arts and Technology, University of California, Santa Barbara.
2021 - 2024	M.S. in Computer Science, University of California, Santa Barbara.
2008 - 2012	B.A. in Journalism, Communication University of China.

Publications

Xu, C., Kumaran, R., Stier, N., Yu, K., and Höllerer, T., Multimodal 3D Fusion and In-Situ Learning for Spatially Aware AI, under review, 2024.

Xu, C., Lien, K.C., and Höllerer, T., Comparing Zealous and Restrained AI Recommendations in High-stakes Human-AI Collaboration, *ACM Conference on Human Factors in Computing Systems (CHI), 2023.*

Zhu, J., Kumaran, R., **Xu, C.**, and Höllerer, T., Free-form Conversation with Human and Symbolic Avatars in Mixed Reality, *IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2023.*

Xu, C., Dong, B., Stier, N., McCully, C., Howell, D. A., Sen, P., and Höllerer, T., Interactive Segmentation and Visualization for Tiny Objects in Multi-megapixel Images, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, demo track.*

Xu, C., McCully, C., Dong, B., Howell, D. A., and Sen, P., Cosmic-CoNN: A Cosmic Ray Detection Deep Learning Framework, Dataset, and Toolbox, *240th American Astronomical Society meeting, oral. The Astrophysical Journal, Volume 942, Number 2.*

Hiramatsu, D. et al., including **Xu, C.** The electron capture origin of supernova 2018zd, *Nature Astronomy, Volume 5, Issue 9.*

Work Experience

Summer 2023	Research Scientist/Engineer Intern, Adobe
Jan. - Sept. 2022	Computer Vision Intern, Appen
Summer 2021	Computer Vision Researcher, Benioff Ocean Science Laboratory
Summer 2019	Imaging Intern, Las Cumbres Observatory
2015 - 2016	Multimedia Producer, BBC News
2012 - 2015	Video Journalist, CNN International

Abstract

Understanding and Facilitating Human-AI Teaming for Real-World Computer Vision Tasks

by

Chengyuan Xu 许程远

Recent machine learning research has demonstrated that many task-specific AI models now reach or surpass human performance on static benchmarks. However, in real-world applications where human users collaborate with, or rely on AIs, key questions remain: Do these advancements in AI models inherently improve the user experience or augment users' capabilities? When and how should we partner users with AI to form effective human-AI teams? This dissertation explores new forms of human-AI collaboration in the context of real-world computer vision tasks. We demonstrate different user roles in diverse AI-assisted workflows – from passive recipients of AI model outputs to active participants who steer the shaping of the model. 1) We developed intuitive user interfaces to make deep learning accessible to end users, in this case astrophysicists, without requiring knowledge in machine learning. The end-to-end model enhances the accuracy of automated processing of daily space observations from 20+ telescopes globally. The streamlined interface injects confidence into researchers' AI-supported analysis of scientific imagery. 2) We proposed the concept of "restrained and zealous AIs" to harness the complementary strength in human-AI teams. Insights from a month-long user study involving 78 professional data annotators suggest that recommendations from ill-suited AI counterparts may detrimentally affect users' skills. 3) Finally, we brought a novel concept of "in-situ learning" to augmented reality, where the user interacts with physical objects to train spatially-aware AI models that can remember the personalized environment and

objects for various tasks. Each project brings the end user to a more active and engaged role in the inference, training, and evaluation processes of human-in-the-loop machine learning. In summary, this dissertation provides insights into good practices for teaming humans with AI for real-world collaboration, informing the design of future AI-assisted systems.

Contents

Curriculum Vitae **vi**

Abstract **vii**

1 Introduction **1**
1.1 Why study human-AI teaming 1
1.2 From end-to-end training to interactive learning 4
1.3 Teaming opportunities 11

2 End-to-End Models for Autonomous Image Processing **17**
2.1 Introduction 19
2.2 LCO CR Dataset 22
2.3 Deep-learning framework 25
2.4 Results 34
2.5 Toolkit 41
2.6 Conclusion 44
2.7 Acknowledgments 45
2.8 Appendix 46

3 Human-in-the-loop Image Processing **52**
3.1 Introduction 54
3.2 Usage 58
3.3 System Design 59
3.4 Advantages Over Existing Tools 61
3.5 Discussion 63

4 Forming Human-AI Teams for High-Stakes Tasks **65**
4.1 Introduction 67
4.2 Related work 72
4.3 Algorithm choices and pilot studies 75
4.4 Experiments 81
4.5 Results 85

4.6 Discussion 97
4.7 Conclusion 102
4.8 Acknowledgments 103

5 In-Situ Learning for User-Guided Personal AI 104
5.1 Motivations for in-situ machine learning 105
5.2 In-situ learning proof-of-concept prototypes 108
5.3 Multimodal 3D Fusion and In-Situ Learning for Spatially Aware AI . . . 115
5.4 Implications of in-situ learning 143

6 Summary and Discussion 145
6.1 Summary of contributions 145
6.2 Human-AI teaming for creative applications 148
6.3 Human-AI teaming and the growing model size 150

Bibliography 153

Chapter 1

Introduction

1.1 Why study human-AI teaming

Computers are good at fast, complex, and repeated computations. When sufficient data and computational power are available, machine learning (ML), or artificial intelligence (AI), is becoming a pervasive solution for many well-formulated problems. On the other hand, humans have the intelligence to recognize hard-to-formulate patterns with just a small amount of data, solving complex problems with their human heuristics, life experience, or domain knowledge. Humans can also provide additional information or real-time feedback to correct machine mistakes, customize AI models, and guide a system to behave according to the user's personal preferences or immediate needs. These observations naturally point to human-AI teaming, a collaborative partnership that can provide better team performance or user experience than either party working alone.

Human-AI teaming has been a vision for many years. Licklider's 1960 Man-Computer Symbiosis [1] predicted the very close coupling between humans and computers "to cooperate in making decisions and controlling complex situations". Since then, this vision has gradually taken shape. Sam Altman, the leading figure behind the later revolutionary

ChatGPT, commented in 2014 that "Computers and humans are very good at very different things. A computer doctor will out-crunch the numbers and do a better job than a human on looking at massive amounts of data, but on cases that require judgments, creativity, or empathy, we are nowhere near any computer system that is any good at this"[1]. Recent huge leaps in computer vision foundation models, large language models (LLMs), and generative AI made us reconsider our assumptions about the limitations of AI in various human-like processes, reshaping our perspectives and expectations.

Human-AI teaming aims to harness the advantages of both while at the same time overcoming their respective limitations. The collective intelligence has been shown to improve the clinical diagnostic accuracy in cases of pneumonia [2] and metastatic breast cancer [3], assist our driving [4], increase efficiency in crowdsourcing [5] and large-scale citizens' science projects like Galaxy Zoo [6]. Additionally, it assists in decision making in many relatively low-stakes decisions such as writing suggestions[2], moderates social media comments [7], improves better game plays [8], and even supports more robust business decisions [9].

There is no doubt that AI models will be more versatile, powerful, and accurate, but it is also clear that humans should remain in control, especially in the decision-making process for high-stakes tasks. The question that remains, which is also the central theme of this dissertation, is how to make AI work better for human users by forming effective human-AI teams. In this dissertation, we shape a research space where users play different roles in diverse AI-assisted workflows – from passive recipients of AI model outputs to active participants who steer the shaping of the model through four concrete topics that contribute to the overarching theme in the following chapters:

1) End-to-End Models for Autonomous Image Processing

[1]WSJ Tech Live 2023: https://www.youtube.com/watch?v=byYlC2cagLw
[2]Grammarly https://www.grammarly.com/

2) Human-in-the-loop Image Processing

3) Forming Human-AI Teams for High-Stakes Tasks

4) In-Situ Learning for User-Guided Personal AI

From an end-to-end model for autonomous image processing to fully user-guided personal AIs through in-situ learning, the research projects discussed in this thesis naturally formed a progressive path in which each project elevates the end user to a more active and engaged role in the inference, training, and evaluation processes of human-in-the-loop machine learning, gradually deepening the level of collaboration between the user and AI models.

Thus this dissertation reflects my journey of understanding human-AI teaming in real-world computer vision tasks, as well as my contributions to enabling teaming through various research projects carried out during my Ph.D. studies. I believe the following background information is useful to help readers better understand the works that we will discuss in this thesis:

• The projects in this thesis provide computer vision and human-AI collaboration solutions across various real-world tasks, many of which result from partnerships with industry or research organizations that are keen to solve challenging problems with AI.

• Each project adds uniquely to the overarching theme, offering contributions such as new datasets, AI models, tools, or novel interfaces that facilitate human-AI collaboration and the interactive training of machine learning systems.

• Throughout the various chapters of this thesis, we make reference to users with diverse backgrounds, ranging from domain experts such as astrophysicists, machine learning practitioners, and human-computer interaction (HCI) researchers, to the general public like users of augmented reality (AR) headsets.

• The projects presented in this thesis, while diverse in their contributions, are unified in their use of computer vision as the underlying technological discipline, encompassing

tasks such as semantic segmentation, object detection, classification, 3D reconstruction, and scene understanding.

1.2 From end-to-end training to interactive learning

Building end-to-end machine learning models is a valuable contribution to computer vision (CV) research. They are particularly useful for fully autonomous CV tasks that require high throughput, such as detecting product defects in a production line, or continuous observation and alerts, such as security cameras. From the end user's perspective, i.e., when the model is deployed and used for inference, an end-to-end model requires minimal manual input for pre-processing or post-processing, making the AI model more accessible to users who do not have machine learning knowledge and easier integration into a larger system. The model can be seen as a black box that takes raw data, such as images, as input and outputs predictions that are ready for downstream tasks or decision-making. These predictions can include classification (e.g., identifying dog images), detection (e.g., locating a dog in an image by bounding box), segmentation masks (e.g., labeling pixels that belong to the dog), and more. We visualize a few of such examples in Figure 1.1.

While end-to-end ML models are incredibly powerful in certain use cases, it also means that little or no human input or feedback is taken into account during both the model training and the inference processes. Conventionally, ML experts and engineers train AI models in a rather automatic workflow. When models are in use, the end users hardly play any active role other than passively waiting for the model to produce the results. These predictions can sometimes be unexpected in real-world scenarios. Many factors can lead to surprising, especially failing results: insufficient training data, overfitted models, out-of-distribution inference data, etc.

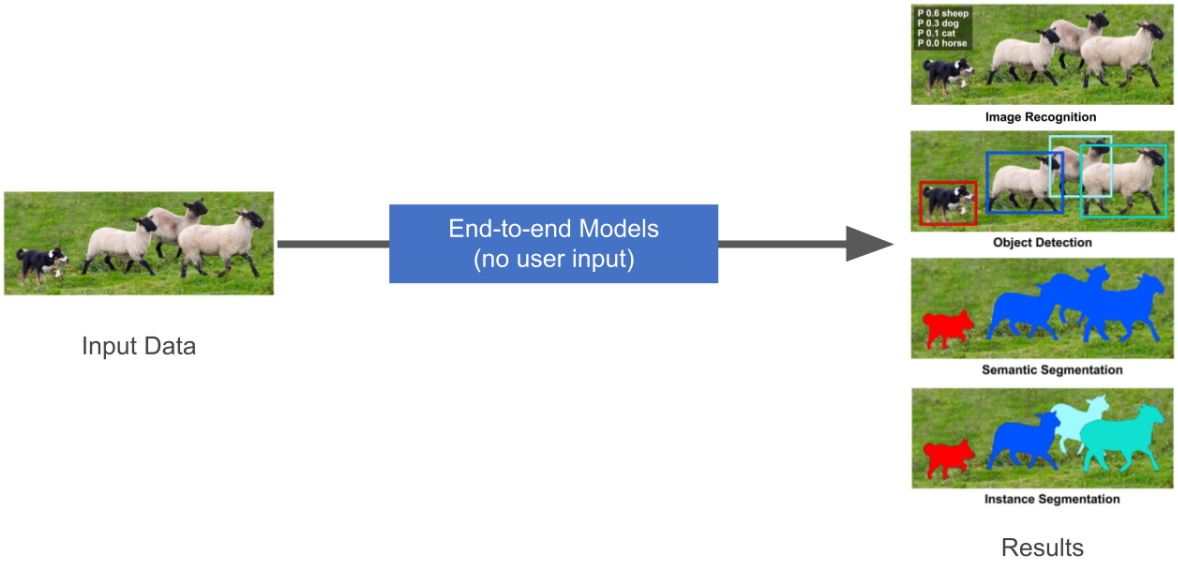


Figure 1.1: Examples of classic computer vision tasks [10]. For end users and downstream tasks, end-to-end models can be treated as a black box that produces the result but provides end users with little knowledge or control to steer, update, or improve the model's behavior.

The shaping of a typical machine learning model can generally be divided into four main stages: 1) preparing data and models, 2) training the model, 3) evaluating model performance, and 4) deploying the model, with specific tasks within each stage [11], as illustrated in Figure 1.2. In conventional ML research and development, the workflow requires one or more ML experts' efforts, but end users rarely participate in this process. For instance, a customer who uses a photo editing app that recognizes faces for virtual makeup or detects persons for bokeh effects is guaranteed that the newly taken photograph was never used to train the face/person segmentation model.

Specifically, in the context of data-driven machine learning, "data collection" and "feature selection" are the most critical tasks that determine the feasibility of an ML model. If the end users are domain experts whose domain-specific knowledge is necessary to define the model's objective, This early stage is an opportunity for such end users

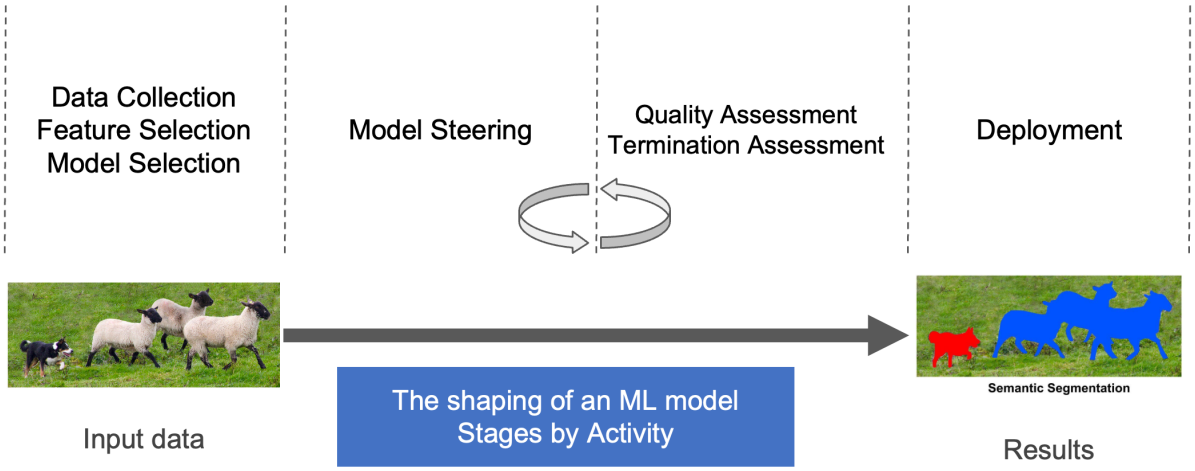


Figure 1.2: From left to right, it generally takes four main stages to design, train, evaluate, and deliver a conventional end-to-end machine learning model. Within each stage, there are multiple specific tasks that involve ML specialists, but rarely can the model's end users participate in the process.

to get involved by providing task specifics and training data in collaboration with ML experts [12]. Another opportunity for domain experts' involvement is in the later stages by providing feedback on the model's performance evaluation. Once the model architecture, the objective function (loss function), and the training strategy (supervised, unsupervised, self-supervised, etc.) are confirmed, the "model steering", "quality assessment", and "termination assessment" stages, an iterative process that is generally referred to as "model training", are determined by the data and carried out automatically. An algorithm or an ML expert assesses the evaluation results periodically to determine if and when to stop the training.

From the human-computer interaction perspective, it's a missed opportunity not to leverage human insights and user guidance to correct the model's mistakes in exceptional cases. If appropriately designed, user input can improve the AI model's performance (benchmark evaluation) or the collaborative human-AI team's performance (user evaluation) in an interactive fashion. When a large annotated dataset is not available or hard to collect, input and feedback from the users become even more valuable when building

AI models for collaboration.

Given the aforementioned constraints of end-to-end model training, researchers saw opportunities to introduce user input and feedback into a new learning paradigm and proposed interactive machine learning (IML). Ware et al. [13] showed that training models in an interactive fashion have the potential to produce accurate classifiers in the hands of a domain expert with help from an interface that improves domain knowledge integration into the model-building process. Fails and Olsen [14] demonstrated the IML model (Figure 1.3) with a more visually appealing application – image segmentation to show that IML can help designers, users need not have profound knowledge in machine learning or image processing, to incorporate intelligence into perceptual interface designs. As illustrated in Figure 1.4, users were able to directly draw on the image to provide the training data, the ground truth, as well as the user's objective to the model; in the meantime, they also evaluate the system's current state using predictions shown on the same image and provide guidance for improvement in an iterative fashion. The simple and intuitive interface insulates the user from knowing the underlying machine learning mechanisms.

These early efforts formulated the basic interactions and collaborations paradigm between humans and the IML system, which Dudley and Kristensson [11] illustrated in a structural breakdown of four key components in Figure 1.5: the user, the data, the model, and the interface. The user is the main driving force of the iterative learning process. A user's interaction with the model may be implicit (modifying data to update the model) or explicit (directly modifying the model parameters), and sometimes both, depending on the application setting.

The change of end user's role in the AI model's training and inference process marks the core difference behind interactive machine learning in contrast to the end-to-end training we discussed earlier. We showed that the general

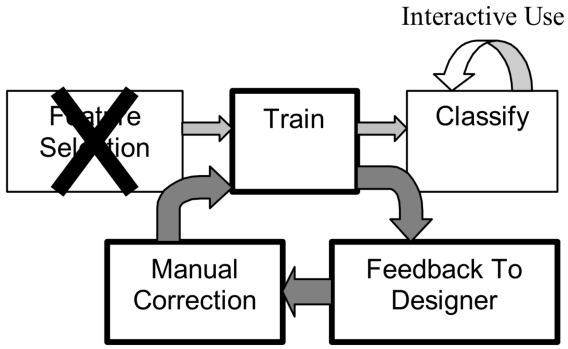


Figure 1.3: The interactive machine learning (IML) model proposed by [14]. In contrast to the conventional machine learning training that we showed in Figure 1.2, Fails and Olsen replaced the Feature Selection in static data pre-processing with users' manual correction as dynamic input to feature rapid "train-feedback-correct" cycles.

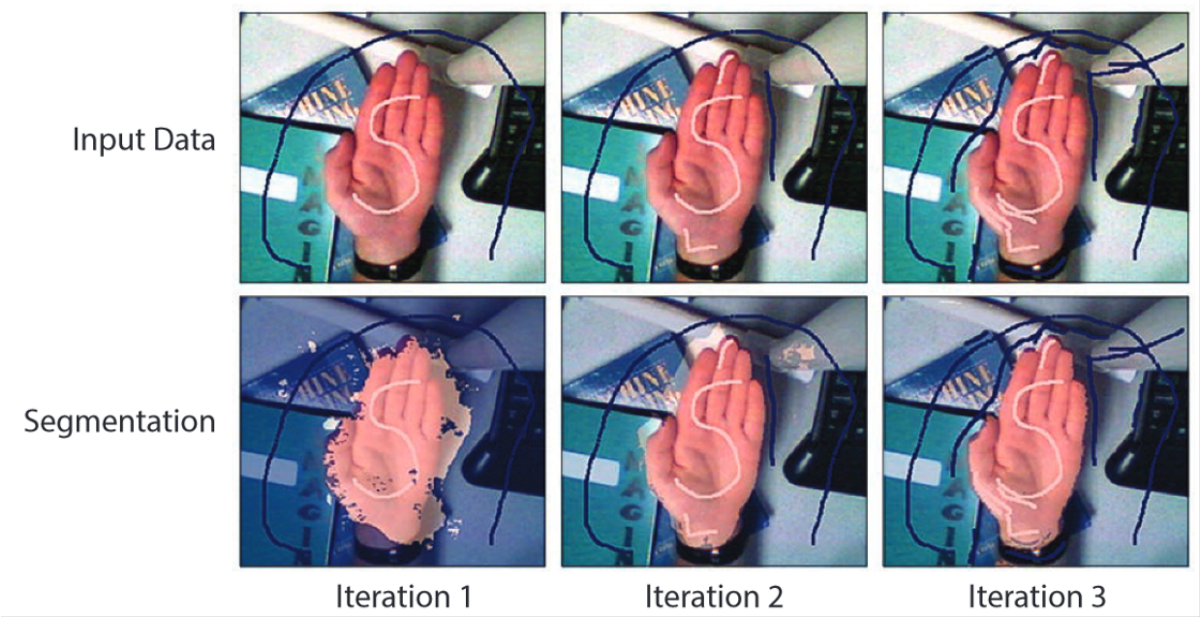


Figure 1.4: The process of creating a classifier using Crayons [14].

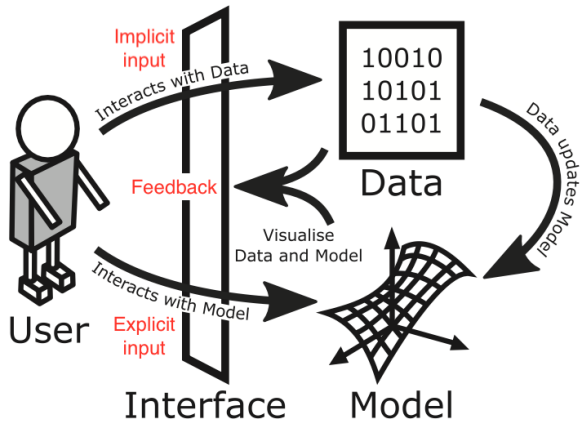


Figure 1.5: Structural breakdown of a generic interactive machine learning system [11] (with our edits in red).

public users have nearly zero input in the shaping of conventional machine learning models, which are largely carried out by skilled ML practitioners. In applied ML workflows involving specialized domains, even the domain expert end users often have limited involvement in the model training process. They often relied on skilled ML practitioners because of the intricacies of building ML systems. A domain expert's inputs are often limited to providing data, answering domain-related questions, and giving feedback about the learned model [12]. In the meantime, ML practitioners try to "understand and translate" the end user's intention into equations, code, and eventually a model, with the risk of a less optimal design due to the lack of expertise in the application domain.

IML approaches model training by placing the end user, along with their domain knowledge, at the center of the stage. By shielding end users from the algorithms behind the machine learning system, domain experts or end users can focus on the application task – providing feedback and steering model development through an interface, all without the need for an in-depth understanding of the technical details in machine learning.

Interactive machine learning shares some core values in human-centered AI (HCAI), AI systems that amplify and augment rather than displace human abilities. As an um-

brella of specific learning approaches, IML and HCAI both seek to preserve human control in a way that ensures AI meets our needs while also operating transparently, delivering equitable outcomes, and respecting privacy [3].

In an ideal IML workflow, humans (especially end users) play a central and active role by providing training data (e.g., domain knowledge or user preference) to the model, reviewing the model's current state, and taking appropriate action (e.g., more training samples or positive/negative feedback) in an iterative fashion. The benefits of such collaboration are even more evident in situations in which highly personalized AI models are preferred – physicians or astrophysicists who have specialized data that are not trained by mainstream AI models, or AR users who work in diverse personal spaces where the generalized scene understanding models perform poorly. In our time of exponentially increasing volume of data, such scenarios are becoming more common as machine learning is being applied in many aspects of society and more disciplines of scientific research for problem-solving.

A wide range of ML applications have taken advantage of the ideas behind interactive machine learning. Modern recommender systems harvest data through user interactions. For instance, TikTok learns highly customized video flows that fit users' personal likings based on their proactive tagging like "heart" or passive cues like time spent on different types of videos (each video has been fully analyzed to represent many sophisticated quantitative features). Popular social media platforms have replaced the conventional time-based news feed with "smart" recommendations based on algorithms that maximize user activity. The key difference that makes this type of user participation different from the focus of this dissertation is that in these products, end users often contribute to the model-building process unknowingly, while the human-AI teaming we propose requires users to actively drive the AI's learning process.

[3] https://research.ibm.com/blog/what-is-human-centered-ai

1.3 Teaming opportunities

A dive into the concrete construction process of a machine learning model can help identify at which stage ML experts or end users can participate in an AI-assisted workflow or influence the shaping of the model, i.e., the human-AI teaming opportunities. Dudley and Kristensson [11] proposed a generalized workflow of interactive machine learning models based on distinct user activities and interactions, shown in Figure 1.6. We extend the workflow with an extra task of "data collection" at the very beginning since multiple projects in this dissertation contributed on this front. The workflow is helpful in anchoring user participation to seven concrete activities. As we already did in Figure 1.2, we can map the seven activities to the following stages:

Preparation: Data Collection, Feature Selection, Model Selection

Model Training: Model Steering, Quality Assessment, Termination Assessment

Deployment: Transfer

In Figure 1.7 and Figure 1.8, we visualize how each chapter in this dissertation contributes to the different activities or stages of the IML workflow and human-AI teaming. Specifically:

Chapter 2 In this chapter, we present the Cosmic-CoNN framework for accurate detection of cosmic rays (CRs) in astronomical images, showcasing the power of deep learning to address a long-time challenge in astrophysics. By building a large and diverse CR imagery dataset from Las Cumbres Observatory, along with innovations in network architecture and loss functions, this framework not only achieves high precision and recall in CR detection but also demonstrates robustness across new unseen telescopes and imaging conditions. Cosmic-CoNN's main contributions lie in high accuracy and robust autonomous processing. The framework's design and implementation pave the way for more interactive and collaborative models of human-AI interaction in scientific

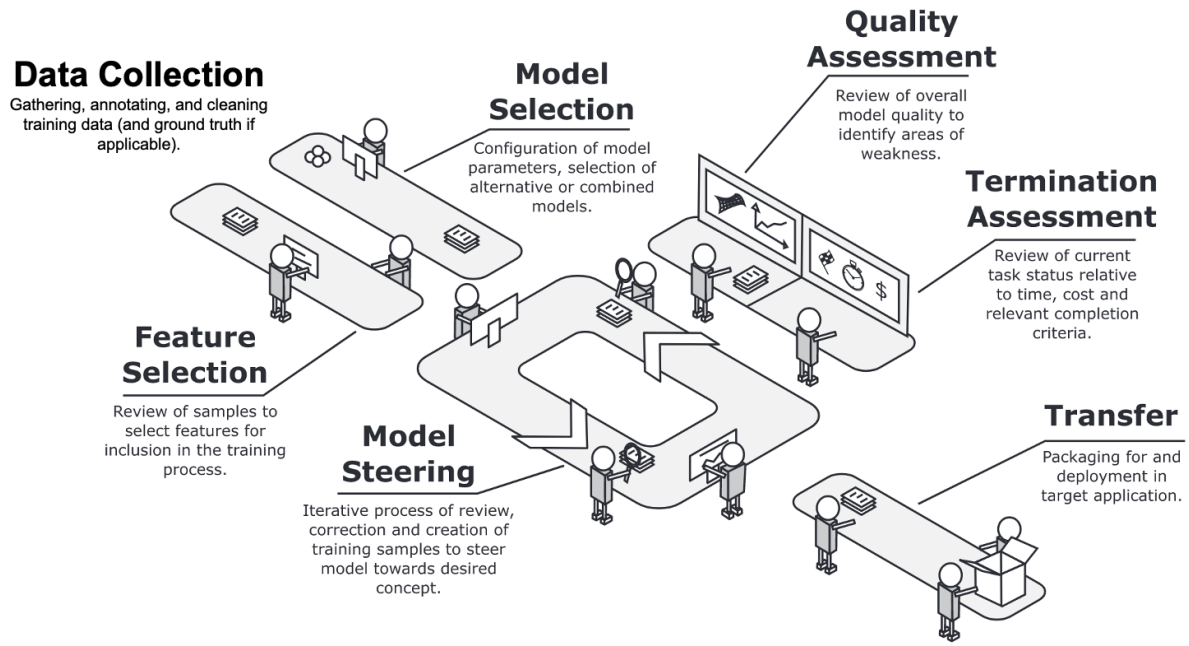


Figure 1.6: A generalized interactive machine learning workflow as a behavioral breakdown in distinct user activities [11].

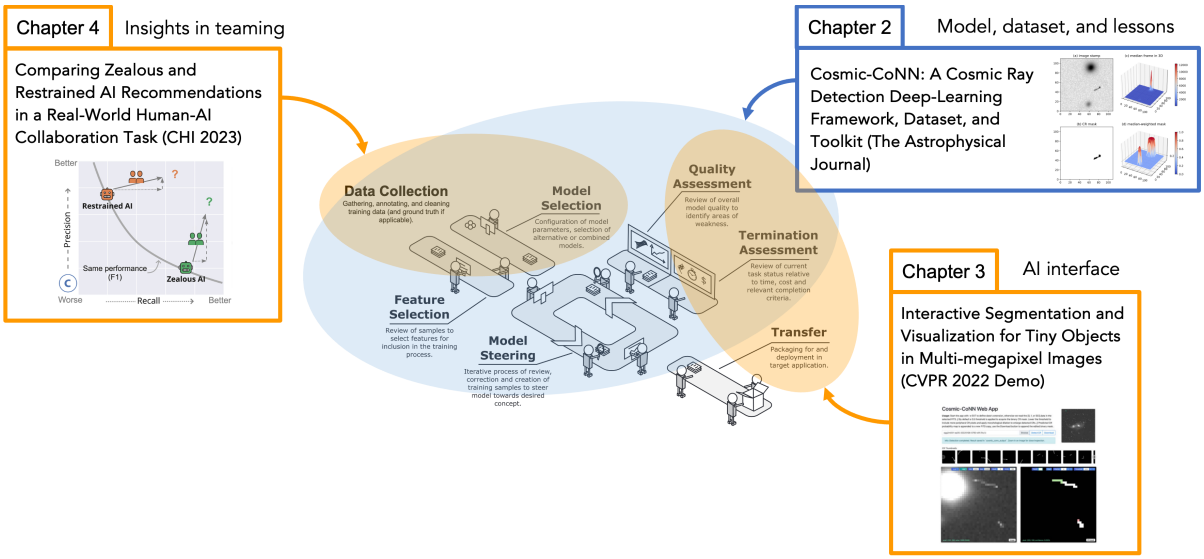


Figure 1.7: Based on the IML workflow model proposed by Dudley and Kristensson [11], we align Chapters 2, 3, and 4 with specific activities that each work contributes to in the model construction process.

research.

This project, while initially focused on autonomous computer vision in a scientific context, sets the stage for exploring real-world AI applications where domain expertise plays a critical role. The insights gained from working closely with astrophysicists during the development of Cosmic-CoNN have highlighted a gap between AI capabilities and user engagement. Future efforts could explore interfaces that allow scientists to adjust parameters, influence training processes, or interactively refine AI predictions, moving towards a human-AI team that enhances both the AI's utility and the scientific insights it can generate.

Chapter 3 This chapter introduces an interactive segmentation and visualization toolkit designed to address the specific challenges associated with detecting minuscule contaminated pixels in large, multi-megapixel high-dynamic-range (HDR) images, in this case, cosmic ray (CR) detection in astronomical images. The toolkit integrates model inference, HDR visualization, and segmentation mask inspection and editing within a single, user-friendly graphical user interface (GUI). By consolidating these processes into a streamlined interface, the toolkit democratizes computer vision model's capabilities and simplifies the scientific imagery analysis workflow, thus facilitating more accurate scientific interpretation of imagery data.

The toolkit exemplifies how advanced computer vision technologies can be made accessible to domain experts without requiring deep technical knowledge in machine learning, by providing interactive tools that allow users to adjust and refine AI-generated segmentation masks. The toolkit supports a human-in-the-loop approach, enabling users to directly impact the AI's behavior. This human-in-the-loop collaboration enhances the black-box AI's utility while also allowing for human expertise to play a crucial role, ingesting confidence into the user's interpretation of the AI's behavior and fostering an effective partnership between human users and AI-assisted systems in the field of

astronomical research and beyond.

Chapter 4 In this chapter, we investigate the impact of AI assistance in human-AI collaboration, focusing particularly on how variations in AI recommendations' recall and precision influence human performance and team performance in high-stakes, recall-demanding tasks. The research introduces a pair of novel AI concepts: a "restrained" AI designed for high-precision recommendations, and a "zealous" AI optimized for high recall. We conducted a large-scale, month-long user study to assess their impact on 78 professional annotators engaged in a video anonymization task. Insights from analyzing over 3,000 person-hours of annotation work demonstrate that carefully designing an AI system's attributes while considering the complementary strengths of humans and AI can effectively enhance the overall team performance. We further found that collaborating with an ill-suited AI teammate for just a couple of weeks can detrimentally affect user skills if annotators lose AI support again, highlighting the importance of aligning AI capabilities with user strength and the demands of the task at hand.

The findings of this study contribute to improving human-AI teaming by illustrating that the interaction between human cognitive abilities and AI-assisted tools is complex and deeply affected by the specific design of the AI system. Particularly, the results indicate that AI systems designed with an understanding of the task's priority and the inherent strengths and weaknesses of human collaborators can lead to improved team performance. When AI assistance is later withdrawn, the observations around negative training effects provide crucial insights for designing sustainable human-AI collaborative environments that augment rather than diminish human skills over time. This research complements the dissertation's theme by providing a concrete example of how AI design and deployment in teamwork settings can significantly influence human-AI teaming, emphasizing the need for thoughtful integration of AI systems into human-centric workflows.

Chapter 5 Finally, combining everything we have learned, we facilitate real-world

human-AI teaming by proposing a novel concept of "in-situ" machine learning, based on the core idea of encoding real-time user-collected data and feedback into a neural network, such that the network itself serves as both the knowledge container and decision-making unit for downstream tasks. We demonstrate the effectiveness of in-situ learning through three diverse use cases: 1) A pose estimation model that learns a user's unique poses in less than a minute and can be used as a personalized physical therapy trainer; 2) Flexible segmentation models that can learn user-defined abstract concepts in both 2D and 3D with simple strokes as model guidance; and 3) A full-fledged augmented reality system that utilizes in-situ learning for physical environments learning. The user interacts with physical objects to train spatially-aware AI models that can remember the personalized environment and objects in real time. In conjunction with a custom 3D reconstruction pipeline that infuses neural vision-language features into the 3D representations of the environments and individual objects, users achieved the highest level of interactive learning – throughout their AR interaction via the AR headset interface, they provide the training data and select the important features, assess the model's quality to decide if more samples or training are needed, and utilize the personalized AI that is specifically trained for their own spaces for tasks like tracking the object changes.

In summary, each project featured in the described chapters plays a crucial role in enhancing user involvement in human-in-the-loop machine learning processes. By elevating the end user to a more active and engaged role in the core stages of inference, training, and evaluation, these projects contribute to the advancement of human-AI teaming. Higher levels of collaboration not only improve the utility and performance of AI applications but also foster a deeper understanding of human user's collaboration with AI-assisted systems. Ultimately, this leads to more robust, effective, and user-centered AI solutions across various domains, providing insights for future human-AI teaming research and application.

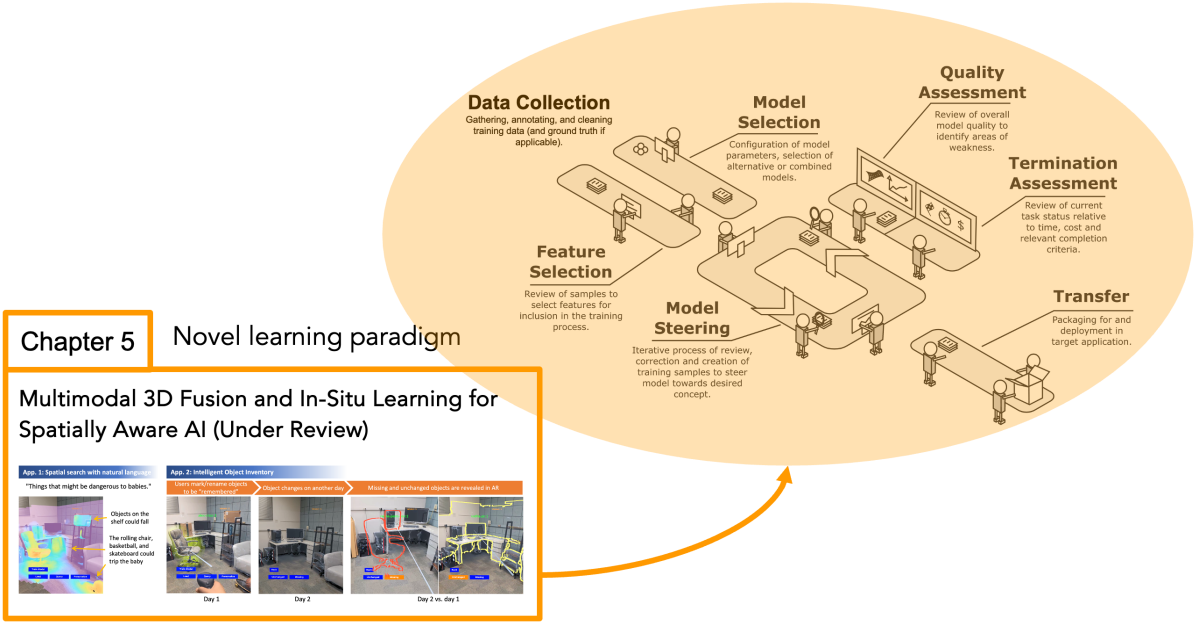


Figure 1.8: Chapters 5 pushes human-AI teaming to the next level as the proposed "in-situ" learning enabled end users to participate in all key activities in the AI model construction process except for model selection. We will discuss in detail the user's role in data collection, guiding the AI training, and quality assessment in this chapter.

Chapter 2

End-to-End Models for Autonomous Image Processing

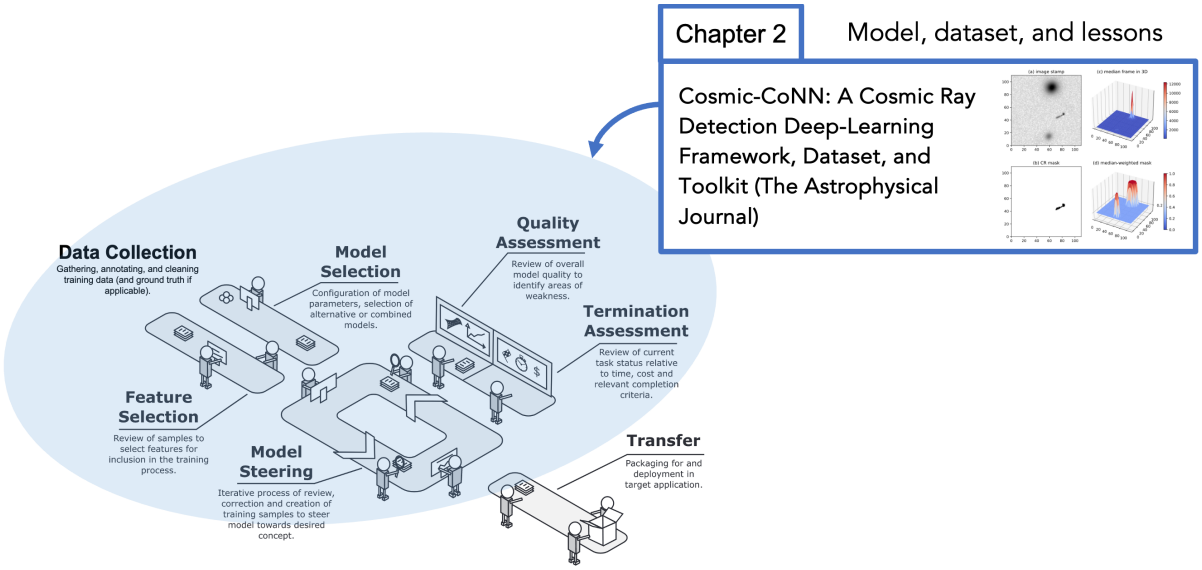


Figure 2.1: This chapter's project, Cosmic-CoNN [15], is a real-world AI for science application that provides a large-scale dataset, a novel lost function that improves the training for space observations, and evaluation benchmarks for cosmic ray detection in ground-based telescopes. Thus, discussions in this chapter touched upon most of the above human activities in shaping a model while no end-user input was considered in the end-to-end model.

We introduce Cosmic-CoNN [15], a deep learning framework that features a breakthrough in both the accuracy and robustness of autonomous cosmic ray artifacts detection in astronomical telescope observations. As a real-world computer vision application that is designed for autonomous batch image processing, the model's end users, in this case, astrophysicists, are passive recipients of the AI model's result with limited control over the model's behavior. However, our close collaboration with astrophysicists has brought a profound understanding of scientific AI applications where domain expertise plays a critical role. It contributes to the vision of human-AI teaming by highlighting the gap between AI capabilities and user engagement in real-world applied ML workflows.

Cosmic-CoNN: A Cosmic-Ray Detection Deep-learning Framework, Data Set, and Toolkit[1]

Rejecting cosmic rays (CRs) is essential for the scientific interpretation of CCD-captured data, but detecting CRs in single-exposure images has remained challenging. Conventional CR detectors require experimental parameter tuning for different instruments, and recent deep learning methods only produce instrument-specific models that suffer from performance loss on telescopes not included in the training data. We present Cosmic-CoNN, a generic CR detector deployed for 24 telescopes at the Las Cumbres Observatory, which is made possible by the three contributions in this work: 1) We build a large and diverse ground-based CR dataset leveraging thousands of images from a global telescope network. 2) We propose a novel loss function and a neural network optimized for telescope imaging data to train generic CR detection models. At 95% recall, our model

[1]The contents of this chapter have been previously published in **The Astrophysical Journal** [15]: C. Xu, C. McCully, B. Dong, D. A. Howell, and P. Sen, "Cosmic-CoNN: A Cosmic-Ray Detection Deep-learning Framework, Data Set, and Toolkit," ApJ, vol. 942, no. 2, p. 73, Jan. 2023, doi: 10.3847/1538-4357/ac9d91

achieves a precision of 93.70% on Las Cumbres imaging data and maintains a consistent performance on new ground-based instruments never used for training. Specifically, the Cosmic-CoNN model trained on the Las Cumbres CR dataset maintains high precisions of 92.03% and 96.69% on Gemini GMOS-N/S 1x1 and 2x2 binning images, respectively. 3) We build a suite of tools including an interactive CR mask visualization and editing interface, console commands, and Python APIs to make automatic, robust CR detection widely accessible by the community of astronomers. Our dataset, open-source codebase, and trained models are available at `https://github.com/cy-xu/cosmic-conn`.

2.1 Introduction

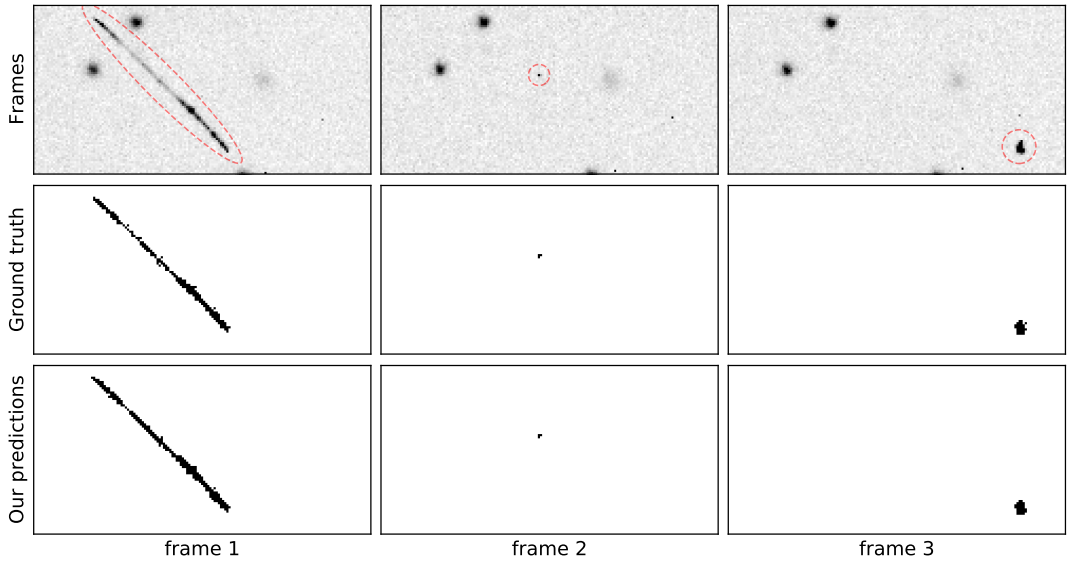


Figure 2.2: Cosmic rays (CRs), labeled with red circles, and other artifacts can be identified by comparing a pixel's deviation from the pixel location's median value in a stack of aligned exposures taken minutes apart. Our deep-learning model predicts a probability map P where $P_{ij} \in [0, 1]$ indicating the likelihood of a pixel being affected by CR using a single frame.

Cosmic rays (CRs) are a key source of artifacts in data from astronomical observations using charge-coupled devices (CCDs). These charged particles excite electrons in the

detector, creating artifacts that can be mistaken for astronomical sources. Space-based instruments like the *Hubble Space Telescope* (*HST*), which are not protected by Earth's atmosphere, are heavily affected by CR, with an average flux density of 0.96 $CR/s/cm^2$ [16]. Ground-based instruments are also affected but at a rate about five orders of magnitude lower, typically of $\sim$ 0.00001 $CR/s/cm^2$ in thin CCDs, as observed in *Las Cumbres Observatory (LCO)* global telescope network imaging data. CCD thickness is another factor that affects an imager's sensitivity to CRs.

Detecting CRs is straightforward when multiple exposures of the same field are available (see example in Fig. 2.2). By comparing the deviation of a pixel from the mean or median value in a stack of aligned images, CRs (and other artifacts) can be effectively identified [17, 18, 19, 20, 21]. However, multiple exposures may not be available, especially for spectroscopic observations. Variations in image quality (e.g., seeing) can also complicate this procedure, so robust detection of CR pixels on individual images is still necessary.

CRs do not travel through the telescope's optical path nor do they follow the point spread function (PSF): they are not blurred by the atmosphere and are therefore sharper than a real PSF. Furthermore, they can come in any incidence angle to have less symmetrical morphologies than real astronomical sources. Several algorithms leverage this feature, like adapted PSF convolution [22], histogram analysis [23], fuzzy logic-based algorithms [24], and Laplacian edge detection [25]. These methods and the IRAF task like `xzap` by M. Dickinson often require adjusting one or more hyper-parameters experimentally to obtain the best result per image. Machine learning algorithms like k-nearest neighbors, multilayer perceptrons [26], and decision-tree classifiers [27] showed promising results on small HST datasets, but lacked generality when compared to image-filtering techniques like `LA Cosmic` [25].

Machine-learning methods have been widely adopted in astronomical research recently

Imager	Class	Pixel Scale ($''$)	Binning	Format (pixels)	Pixel Size (microns)	FOV ($'$)	Filters
SBIG 6303	0.4 m	0.571	1×1	$3K \times 2K$	9	29×29	9
Sinistro	1 m	0.389	1×1	$4K \times 4K$	15	26×26	21
Spectral	2 m	0.304	2×2	$4K \times 4K$	15	10×10	18

Figure 2.3: Table 1: LCO science imagers covered in the CR dataset.

(see [28] for a review). [29] used a convolutional neural network (CNN) to identify CR contaminated pixels in *Hubble Space Telescope (HST) ACS/WFC* images, in a method called `deepCR`. In contrast to using the Laplacian kernel [30] for edge detection as is in LA Cosmic, CNNs learn the intrinsic characteristics of the CR artifacts, enabling it to detect CRs of arbitrary shapes and sizes.

The deepCR model outperforms the state-of-the-art method LA Cosmic without manual parameter tuning, demonstrating the promise of deep learning for CR detection. However, its neural network architecture is an adaptation from U-Net [31] which was originally designed for biomedical imagery tasks that focus on different features than a generic model for astronomical observations from different instruments, specifically ground-based data with variable conditions from multiple instruments. Furthermore, the low CR rates in ground-based data: a ∼1:10,000 ratio between CR and non-CR pixels leads to an extreme class-imbalance issue [32] that provides too few CR pixels for spatial convolution, rendering the training on LCO data more difficult comparing to HST data.

To address these issues, we present Cosmic-CoNN, a deep-learning framework designed to train generic CR detection models for ground-based instruments by explicitly addressing the class-imbalance issue and optimizing the neural network for the astronomical images' unique spatial and numerical features. Cosmic-CoNN also generalizes to other types of data like space-based and spectroscopic observations.

We leverage the publicly available data from *Las Cumbres Observatory (LCO)* to

build a large, diverse CR dataset. *LCO*'s `BANZAI` data pipeline [33] ensures data from different telescopes is not dominated by instrumental signature artifacts. It allows us to label CRs consistently in thousands of observations taken across a wide variety of sites with diverse scientific goals. The LCO CR dataset promises the rich feature coverage required for a generic CR detection model that would work for a variety of ground-based instruments.

This paper is organized as follows: we present the LCO CR dataset in §2.2 and discuss the deep-learning CR-detection framework in §2.3. Extensive evaluations on various types of observations are presented in §2.4. We introduce the toolkit and the software APIs in §2.5, and conclude the paper with a discussion in §2.6.

2.2 LCO CR Dataset

Deep-learning models are data driven. A robust and generic CR-detection model requires a large number of diverse observations from various instruments and the CRs need to be labeled accurately and consistently across different instruments. With this in mind, we build a custom Python CR-labeling pipeline to generate a large cosmic ray ground-truth dataset, leveraging some unique characteristics of *Las Cumbres Observatory* (*LCO*) global telescope network.

Our CR-labeling pipeline stacks consecutive images of the same field to identify cosmic rays. To limit artifacts due to variations in CCD response, we only selected sequences that have at least three repeated observations with identical exposure time and filter. The LCO CR dataset consists of over 4,500 scientific images from *LCO*'s 23 globally distributed telescopes. About half of the images are $4K \times 4K$ pixels resolution and the rest are $3K \times 2K$ or $2K \times 2K$. To the best of our knowledge, this is the largest cosmic ray dataset that identifies CRs in science images across various ground-based instruments.

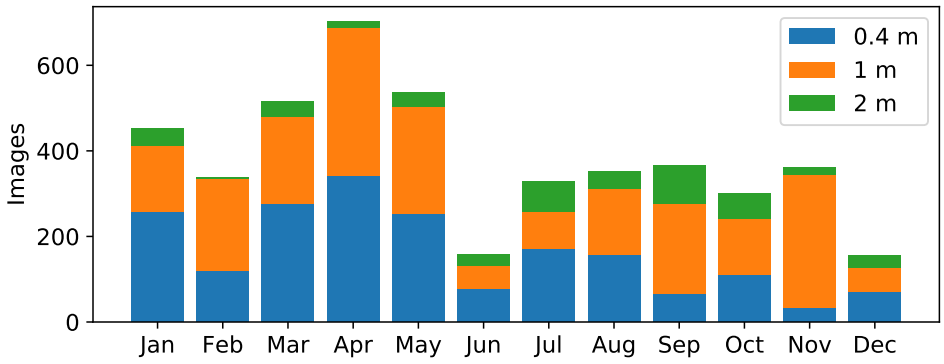


Figure 2.4: LCO CR dataset sample distribution by month and telescope class, from November 2018 to December 2019. Diverse source densities sampled around the year help improve model robustness.

Each sample in our dataset is a multi-extension `FITS` file including three images, the corresponding CR masks, and ignore masks. We encoded hot pixels, pixels with no data, and astronomical sources in the ignore masks to reject false-positive CR pixels. The implementation of our ground-truth CR-labeling pipeline is presented in Appendix 2.8.1. The LCO CR dataset is available for download at `https://zenodo.org/record/5034763`.

The dataset covers a variety of CCD imagers with different pixel scales, field of views, and filters used in *LCO*'s global telescopes network (Table 2.3). From a deep-learning perspective, diverse data greatly benefits model generality. But having ground-truth CRs labeled consistently on different instruments is not a trivial task. The `BANZAI` data reduction pipeline [33] performed instrumental signature removal (bad-pixel masking, bias and dark removal, flat-field correction), making *LCO* data suitable for building such a dataset. Instrument artifacts exist as two identical CCDs could have different response curves after years of bombardment by photons and cosmic rays. The standardized data reduction is key to allow our CR-labeling pipeline to consistently and accurately label CRs across various instruments.

We chose images from across three telescope classes and across the year as shown in Fig. 2.4. Images from different times of the year sampled a variety of source densities for

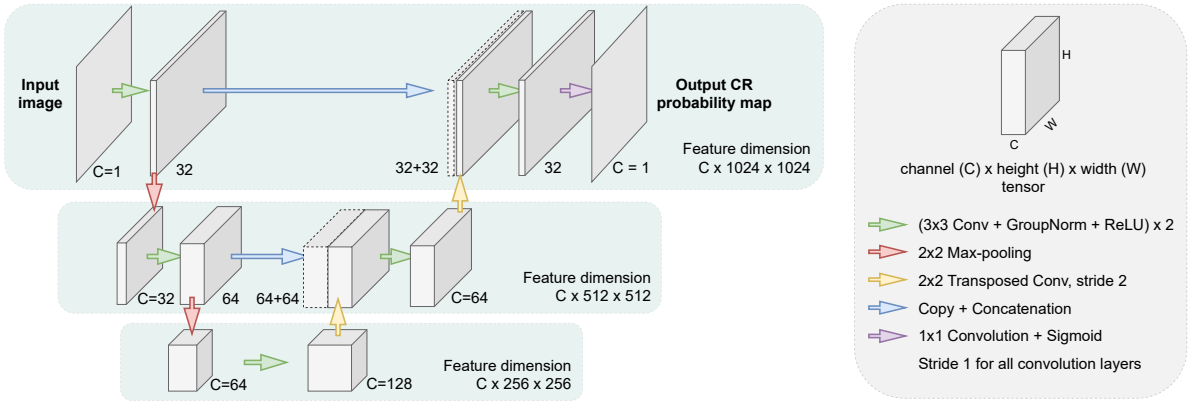


Figure 2.5: Cosmic-CoNN's neural network architecture is based on U-Net. The symmetric design concatenates high-resolution features from the downsampling path to the upsampling path via skip connections (blue arrows), allowing the network to propagate contextual information to higher resolution layers, thereby producing pixel-level classification predictions on CRs of arbitrary shapes and sizes.

different sets of scientific goals. The varying source density proved to be of great importance to robust CR detection [34]. In the task of CR detection, diversified real objects provide rich features for the negative class, which greatly improves model robustness.

We further constrained a sequence of exposures to come from the same scheduling unit: the frames are typically separated by just a few minutes. Repeated exposures in a short period of time help mitigate the PSF variation induced by atmospheric attenuation but PSF wings still cause noticeable false positive labels adjacent sources. We reject CRs that are overlapping with astronomical sources so that variations in the PSF do not create artifacts in the training samples.

Of all CR pixels, 1.21% were rejected in an effort to tackle the PSF-variation-induced artifacts. This trade-off ensures the remaining 98.79% CR pixels are labeled at higher confidence. Therefore, models trained with this dataset focus on distinguishing CRs from real sources, and it is anticipated that CRs overlapped with sources will not be detected. Training on raw images with arbitrary PSFs also guarantees consistent performance at inference time. In future versions we will model the PSF explicitly to make sure that we

do not bias our training sample.

Our dataset is not affected by transient sources that evolve at a timescale of hours or longer because of the very tight space between exposures. At this timescale, near-Earth objects (NEOs), satellites, and airplanes could still cause false-positive labels in the stack-based CR masks. Large satellite or airplane trails are rejected by our CR-labeling pipeline automatically. A very small fraction of false-positive labels from NEOs and satellites exist but we have manually verified every single mask to ensure their impact is negligible.

2.3 Deep-learning framework

Cosmic-CoNN's neural network architecture is inspired by the recent success of deepCR [29], a U-Net [31] based deep-learning framework that identifies CR-contaminated pixels in imaging data. In contrast to the unique Laplacian kernel used in LA Cosmic [25], a deep CNN model optimizes millions of kernel parameters during training and outputs a pixel-level probability map directly. The U-shaped architecture (Fig.2.5) convolves the image at multiple scales, creating a larger receptive field in deeper layers of its hierarchical architecture to capture not only CRs' morphological features (edges, corners, or sharpness) but also the contextual features from peripheral pixels, allowing it to predict CRs of arbitrary shapes and sizes.

deepCR demonstrates the promise of using CNN-based model for CR detection on *HST ACS/WFC* observations. However, training on ground-based images exposes a number of network architecture and data-sampling limitations it inherited from the U-Net [31]. First, it is worth noting that U-Net was initially proposed to solve biomedical image segmentation problems. The higher dynamic range and extreme spatial variations found in astronomical images need to be addressed explicitly in order to optimize the

neural network for these special features in astronomical data. In addition, the high CR rates in *HST ACS/WFC* data does not reflect the extreme class-imbalance issue observed in *LCO* imaging data. The low CR rates make it difficult for deepCR to train and converge on the ground-based LCO imaging data.

In deepCR, [29] adopted a two-phase training design to address some of these issues. Assuming correct data statistics are learned in the initial phase, the model freezes feature normalization parameters in the second phase in order to converge. This design works when the inference data shares the same statistics with training data, i.e., an instrument-specific model could be learned. But it works against our goal of a generic CR detection model that works for a wide variety of ground-based instruments with varying data statistics.

Cosmic-CoNN adopted the U-shaped architecture and proposed: (§2.3.1) a novel loss function that specifically addresses the class-imbalance issue, and (§2.3.2) adopted data sampling, augmentation, and feature normalization approaches that are more suitable for ground-based data that work jointly to improve model generality and training efficiency.

2.3.1 Median-weighted loss function

The CR-detection task is in essence a pixel-wise binary classification problem. Our goal is to learn a function f which takes an image I as input and outputs P, the probability map of each pixel being affected by CR:
$P = f(I), P_{ij} \in [0, 1]$, where ij is the pixel coordinate. The user could then apply an appropriate threshold on P to acquire the binary CR mask.

Binary cross entropy (BCE) is commonly used to optimize classification models, which can also be used to calculate the loss between the prediction P and the ground-truth CR

mask Y:

$$\begin{aligned} \mathrm{BCE}(P, Y) = -(Y_{ij} \log(P_{ij}) + \\ (1 - Y_{ij}) \log(1 - P_{ij})) \end{aligned} \tag{2.1}$$

where the ground-truth mask Y is defined as $Y_{ij} = 1$ for CR pixels and $Y_{ij} = 0$ for non-CR pixels. The first term $Y_{ij} \log(P_{ij})$ measures the loss for CR pixels and second term for non-CR pixels. The optimization objective is to minimize their sum to account for both CR and non-CR classes.

The low CR rates in LCO data causes the non-CR loss to dominate the total loss. Training on *LCO* imaging data, the observed losses from the two terms in Equation 2.1 have a ratio of ∼1:6300 (averaged over 10 random experiments), with the second term (non-CR loss) dominating the optimization objective. This verifies the class-balance issue.

Furthermore, background pixels are the culprit for an extra layer of imbalance within the non-CR class. From dark background to bright sources, the non-CR class often covers the image's entire dynamic range (see example in Fig. 2.6a,b). Although both labeled as 0 in Y (Fig. 2.6c), the lopsided numerical difference between background and sources in fact creates two sub-classes within the non-CR class to introduce inconsistency, making the training path even more convoluted.

The class imbalance and the numerical imbalance within the non-CR class are clear indications that we should directly focus on learning to distinguish between CRs and sources. It inspired us to create an adaptive per-pixel weighting factor that prioritizes on CR and source pixels by down-weighting the less useful yet dominant loss from background pixels.

Since we already acquired a sequence of consecutive exposures building the LCO CR dataset, we could use the CR-free median frame (Fig. 2.6b) as an unique ground-truth to separate sources from the background. The brightness variation between different

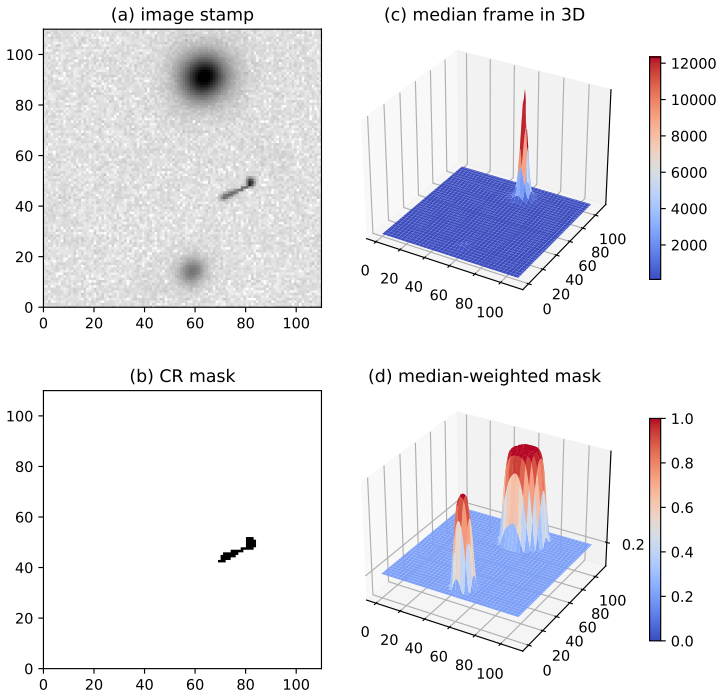


Figure 2.6: 3D visualization of the median-weighted mask. (a) An image stamp that includes sources, CR affected pixels, and background. (b) The ground-truth CR mask shows the imbalance between CR and non-CR pixels. (c) 3D visualization of the CR-free median image shows the non-CR pixels can be further split into two sub--classes: sources and background, while the background pixels may be dominant in quantity. We transform (b) to acquire (d), the median-weighted mask (M) by normalizing the brightness variation between sources. M in Eq. 2.2 adaptively down-weight background pixel loss in the proposed median-weighted loss function. In this figure, $M_{ij} \in [0.2, 1.0]$.

sources makes it hard to use the median frame as a weight mask directly, so we perform a series of transformations (sky subtraction, clipping between one and five robust standard deviations, 5×5 kernel with $\sigma = 2$ Gaussian smoothing, unit normalization, and finally clamping with a lower-bound parameter α) to separate sources from the background to acquire the median-weighted mask (M) shown in Fig. 2.6d. We apply M to the non-CR

loss term in BCE to get the novel median-weighted loss function (L_M):

$$L_M(P, Y, M) = -(Y_{ij} \log(P_{ij}) + \mathbf{M_{ij}}(1 - Y_{ij}) \log(1 - P_{ij})) \tag{2.2}$$

where $M_{ij} \in [\alpha, 1]$. Pixel by pixel, M adaptively down-weights the loss from background by scaling with the lower bound α, mitigating the extreme imbalance between the two loss terms and redefines the optimization objective to directly learning to distinguish between sources and CRs.

With M applied to the second term in BCE, it immediately reduces the observed CR to non-CR class losses to ∼1:300 in Equation 2.2, comparing to the ∼1:6300 using Equation 2.1 (in identical conditions). Although this ratio can be further reduced with a more aggressive weight mask, the median-weighted mask preserves all real sources without introducing inconsistency. After training with 500 images, the observed loss of the two terms further reduce to ∼1:6 using L_M, comparing to ∼1:110 using BCE loss. In Fig. 2.7, we show that the deepCR model optimizes sooner and to a better minimum with L_M while holding other variables constant.

The median-weighted loss function (L_M) makes use of the median frame's unique CR-free property as a robust weighting factor to effectively suppresses the dominating loss from background pixels, at the same time prioritizes on learning to distinguish between CRs and sources by maintaining their weighting factor at 1.0. As training progresses, the lower bound α linearly increases the weight for background pixels from 0.0 to 1.0 so the model could learn a clear boundary for CRs.

We could also cap α at less than 1 to learn a model that produces CR prediction with soft edges, leaving more control to the user-defined threshold when a binary CR mask is needed. We choose to increase α to 1 so that L_M converges to the BCE loss, working

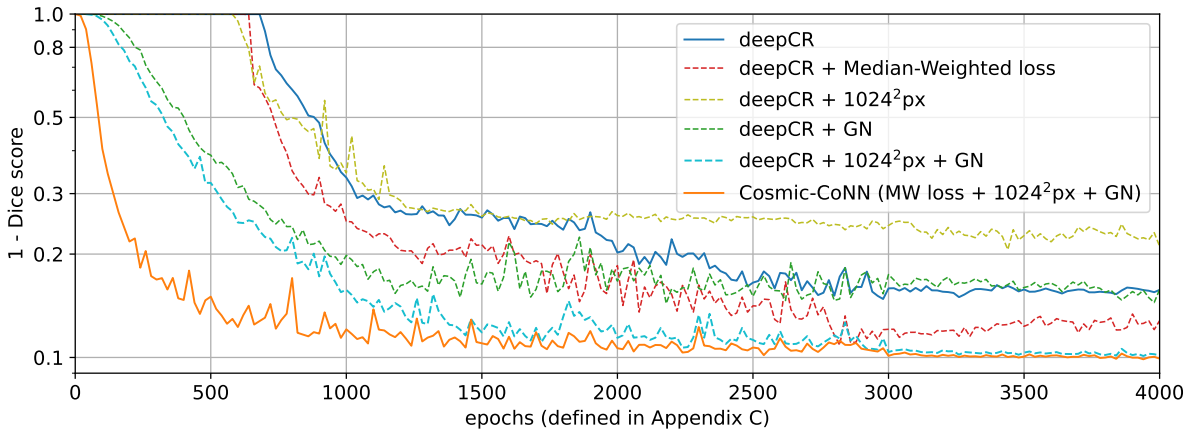


Figure 2.7: Using deepCR as baseline, we demonstrate our proposed improvements' effects to the model performance as a function of training progress. All variant models are initialized with the same random seed, trained on an identical set of LCO data, and evaluated with identical validation images using same model-input dimension. Performance is measured by the Sørensen-Dice coefficient [37] (henceforth, the Dice score) to gauge the similarity between the model's prediction and the ground-truth CR mask. Here we plot (1 - Dice score) in logarithmic scale, lower is better. Models without using group-normalization (GN) were trained in two phases, thus the delayed optimizations that start after 500 epochs. The median-weighted loss help deepCR to achieve better performance, while the larger 1024^2 pixels stamps proved to be vital for models using GN. The proposed median-weighted loss function, increased stamp size, and GN work jointly to allow Cosmic-CoNN to converge rapidly and to a better minimum. Quantitative results are presented in Table. 2.15 in ablation study (Appendix 2.8.2).

with the standard Sigmoid function [35, 36] at the last layer of our network to produce a theoretical best classification boundary of around 0.5. We also experimented using a loss function based on Sørensen-Dice coefficient that is robust for imbalanced data [37] but the model learned a strong bias to avoid CRs near real objects, making the more interpretable BCE-based loss a better choice for optimization.

2.3.2 Data sampling and normalization

Large-scale deep-learning models are often optimized using stochastic gradient descent [38], motivated by stochastic methods' efficiency benefits, at the same time constrained by

the ever-growing dataset size and limited GPU memory (usually on the order of 10 GB) for parallel computation. Model parameters are iteratively optimized over a small batch of data, colloquially known as a mini-batch, randomly sampled from the full dataset. If iterating over all N samples in a dataset is considered an *epoch*, then training a model with n samples in a mini-batch means the model updates about $\lfloor \frac{N}{n} \rfloor$ times in an epoch [39].

By slicing *HST ACS/WFC* images into 256^2 pixel stamps, deepCR [29] samples a mini-batch from a dataset of fixed stamps. However, this approach is unsuitable for ground-based astronomical images featuring much lower CR rates: a small 256^2 stamp might not include a single CR, making many of the samples less useful for training.

Recall that each sample in the LCO CR dataset is a multi-extension `FITS` including three images between $2K \times 2K$ and $4K \times 4K$ pixels. This design empowers a more flexible data-sampling strategy than having the dataset stored in a fixed size. The Cosmic-CoNN framework could crop a stamp of any size, up to the entire image from each `FITS`, ensuring a reasonable number of CRs in every mini-batch. The sparsity of source and CR in ground-based astronomical data motivated us to increase the sampling stamp size to 1024^2 pixels. A larger area is more likely to include all three types of features: sources, CRs, and background in a single stamp and also provides more spatial and contextual information for the convolution operations in CNN models.

One consequence of the increased stamp size is the decreased number of samples in a mini-batch, given the same amount of GPU memory. Increasing the stamp width and height by m times will reduce the batch size n to $\lfloor \frac{n}{m^2} \rfloor$, e.g., the memory that fits a mini-batch of 16×256^2 pixel images can only fit a single 1024^2 pixel image. The accuracy of batch normalization (BN) [40], an important feature-normalization method widely used in deep CNN architectures, including in deepCR, decreases rapidly when the batch size becomes too small, so adopting the proposed larger stamp size alone might even hurt

model accuracy, as shown in Fig. 2.7. We adopt group normalization (GN) [41], whose computation is independent of batch size to address the accuracy loss in BN. Unlike BN which normalizes over all feature channels across all samples in a mini-batch, GN divides feature channels into groups and computes the normalization statistics for each sample. We used GN as a remedy for the decreased batch size but found it playing a major role in improving training efficiency on astronomical imaging data.

The high dynamic range, high variance, low source density, and low CR rates in ground-based astronomical images make it difficult to learn accurate per-sample normalization statistics from small stamps: one sample could include a bright source but another could be entirely dark. By pairing GN with the proposed stamp size of 1024^2 pixels, the learned per-sample normalization is more accurate because of the extra spatial and contextual information from the wider field of view.

As a common practice in deep-learning research, we conduct an ablation study to demonstrate the individual and combined effects of median-weighted loss, 1024^2px sampling size, and GN. The results are presented in Fig. 2.7 and Appendix 2.8.2. Controlled experiments show applying GN alone improves training efficiency but not model performance. By pairing GN with the increased 1024^2 stamps, it dramatically improves performance and model generality, while the proposed new loss function provides Cosmic-CoNN a better convergence path to further improve the model's performance and generality on both *LCO* and *Gemini* instruments (see Table. 2.15).

Finally, in addition to randomly cropping image stamps form a large image, we perform weak data augmentation like random rotations as well as horizontal and vertical mirroring, allowing the model to learn invariance to pose variation in astronomical observations [42]. Strong augmentations like elastic deformations adopted by [31] have proved to be effective to improve performance on a small dataset but we avoided such deformation as it could change real CRs' sharp profiles. Given the large number of diverse samples

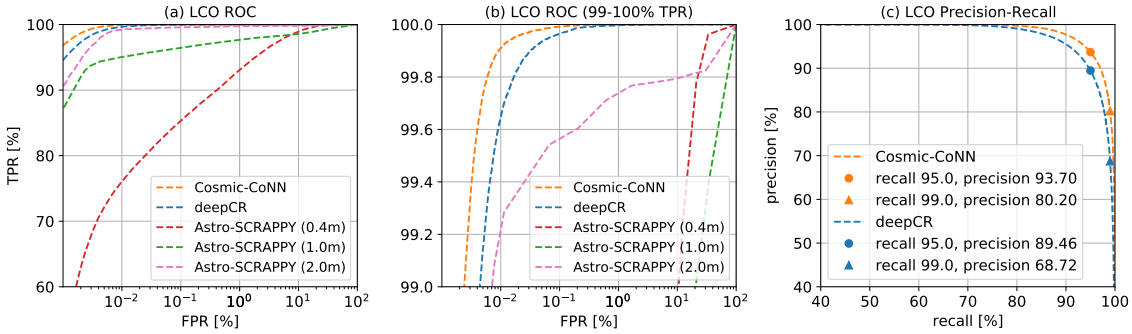


Figure 2.8: Evaluating three CR detectors with ROC and Precision-Recall curves on *LCO* imaging data. It is desirable to have a higher true-positive rate (TPR) at fixed false-positive rates (FPR) in ROC (Equation 2.3,2.4). As illustrated in (a) and (b), Cosmic-CoNN outperforms other methods with higher TPRs overall. The margin of its lead further increases in more strict low FPRs, showing Cosmic-CoNN's robust performance. Circle markers on the Precision-Recall curves in (c) show when 95% of the CR pixels are found (95% recall), Cosmic-CoNN's prediction is over 4% more accurate than deepCR (precision). At 99% recall, Cosmic-CoNN's lead increases to ∼11%.

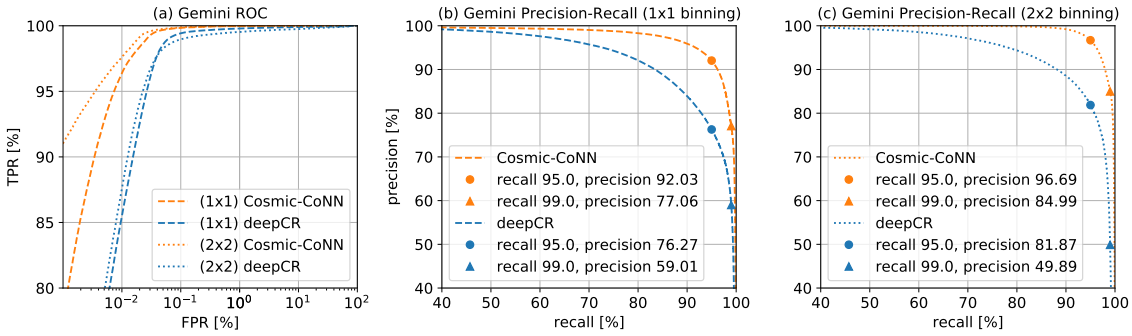


Figure 2.9: We trained both deepCR and Cosmic-CoNN on LCO data and evaluate their performance on new images from previously unseen Gemini GMOS-N/S telescopes. Comparing Gemini data's Precision-Recall curves (Fig.7bc) with LCO's (Fig.6c) shows the Cosmic-CoNN model maintains similar performance while deepCR has visible performance loss (see precision gain/loss in Table. 2.3.2). The consistent performance shows that the Cosmic-CoNN trains a more generic and robust CR detector.

in LCO CR dataset, we found weak augmentations sufficient. With pose augmentation, we also saw more stabilized training and improved performance on *HST ACS/WFC* data, showing that weak augmentation is effective in increasing model robustness.

Method	Test Data	Precision (%) at 95% Recall	TPR (%) at 0.01% FPR	TPR (%) at 0.1% FPR
		(Precision loss/gain on unseen Gemini data)		
Astro-SCRAPPY	LCO Imaging (0m4)	–	76.04	85.17
	LCO Imaging (1m0)	–	95.03	96.41
	LCO Imaging (2m0)	–	99.21	99.56
deepCR (LCO-trained)	LCO Imaging	89.46	99.65	99.97
	GMOS-N/S (1×1 binning)	76.27 *(-13.19)*	85.49	99.43
	GMOS-S (2×2 binning)	81.87 *(-7.59)*	87.58	98.97
Cosmic-CoNN (LCO-trained)	LCO Imaging	**93.70**	**99.91**	**99.99**
	GMOS-N/S (1×1 binning)	**92.03** ***(-1.67)***	96.40	99.84
	GMOS-S (2×2 binning)	**96.69** ***(+2.99)***	97.60	99.89

Figure 2.10: Table 2: We evaluated three CR detection methods on LCO imaging data (§2.4.1). The two deep-learning models deepCR and Cosmic-CoNN trained on LCO images are further evaluated on new images from previously unseen Gemini GMOS-N/S telescopes (§2.4.2), with their relative performance loss on the new instruments indicated in *italic parentheses*. Corresponding to the Precision-Recall curves in Fig. 2.8c and Fig.2.9bc, the Cosmic-CoNN model has little or no performance loss, making it a more generic CR detector for new instruments.

2.4 Results

We trained and evaluated the Cosmic-CoNN framework on various types of instruments and data to access its generalization capabilities. Most importantly, we evaluated the *LCO*-trained model on new imaging data from *Gemini Observatory's GMOS-North/South* telescopes [43] to understand how well the model generalize to other unseen ground-based instruments. The results are presented in the following structure:

- Ground-based imaging data
  - Training and evaluation on LCO data (§4.1)
  - Evaluating LCO-trained models on Gemini GMOS-North/South data (§4.2)
- Space-based imaging data (§4.3)
- Ground-based spectroscopic data (§4.4)

We first use receiver operating characteristic (ROC) curves as an evaluation metric to compare different detectors' performance at varying thresholds. A ROC curve depicts

relative tradeoffs between benefits (true-positive rate, TPR) and costs (false-positive rate, FPR) [44]. In the context of CR detection:

$$\text{TPR} = \frac{\text{CR pixels correctly found}}{\text{All CR pixels}} \tag{2.3}$$

$$\text{FPR} = \frac{\text{Non-CR pixels mistaken as CR}}{\text{All non-CR pixels}}. \tag{2.4}$$

Simply put, a higher TPR is desirable at a fixed FPR. While ROC provides a model-wide evaluation at all possible thresholds, standard ROC can be misleading for datasets that feature different CR rates (e.g., space- vs. ground-based data). Thus it is not suitable to directly compare a model's TPR given the same FPR between different instruments.

The Precision-Recall curve, on the other hand, is a more robust metric for imbalanced datasets [45]. While recall is equivalent to TPR, in the context of CR detection, precision is defined as:

$$\text{Precision} = \frac{\text{CR pixels correctly found}}{\text{All CR pixels predicted by model}}. \tag{2.5}$$

Unlike FPR, precision is determined by the proportion of correct CR predictions given by the model, which is less sensitive to the ratio between CR and non-CR pixels in an image, i.e., it is also less sensitive to the varying CR rates between different datasets. Given a fixed proportion of real CRs correctly discovered (e.g., 95% recall), the better model should make less mistakes, thus a higher precision. It also helps us to understand how well a model performs on two different datasets given the same recall, or vice versa.

The Precision-Recall curve can also be used as an indicator of prediction confidence. We used this property to provide supplementary evidence that helped [46] determine a candidate progenitor to be a new type of stellar explosion – an electron-capture supernova. We rule out the presence of cosmic-ray hits at or around the progenitor site to determine

the peak pixel is an actual stellar PSF with $> 3\sigma$ confidence by plotting deepCR's [29] predicted score on the corresponding Precision-Recall curve.

2.4.1 Training and evaluation on LCO data

For ground-based imaging data, we randomly sampled and withheld ∼10% of images from the LCO CR dataset as the test dataset. We first analyzed the testset using the filtering-based CR detector Astro-SCRAPPY [47] for reference. We used `objlim=2.0` for *LOC* 1.0- and 2.0-meter telescopes' data and `objlim=0.5` for 0.4-meter for optimal performance in different telescope classes. `sigfrac=0.1` is held constant for all telescope classes and we produce the ROC curves by varying the `sigclip` between $[1, 20]$. Both Cosmic-CoNN and deepCR [29] models are trained with identical data and settings. They are evaluated by varying the threshold t. Details of the training environment and experiment settings are presented in Appendix 2.8.3.

The Cosmic-CoNN model achieves 99.91% TPR at a fixed FPR of 0.01%, outperforming other methods, as illustrated in Fig. 2.8a,b. The Precision-Recall curves in Fig. 2.8c shows for both deep-learning models to discover 95% of the real CR pixels (95% recall), the predictions given by Cosmic-CoNN is over 4% more accurate than deepCR's (93.70% vs. 89.46% in Precision). If we continue to lower the threshold to allow 99% of the CR pixels being found, Cosmic-CoNN's lead increases to ∼11%. Quantitative results are presented in Table 2.3.2.

2.4.2 Evaluating LCO-trained models on Gemini GMOS-North/South data

The goal of this work is to produce a generic ground-based CR detection model. In order to understand how well the models trained on LCO CR dataset perform on

unseen instruments, we produced a test dataset consisting of 98 images from the *Gemini Observatory's GMOS North and South* telescopes [43]. The ground-truth CR masks are reduced by the `DRAGONS` software [48] with `hsigma=5.0` to match the setting we used to produce the LCO training data.

As shown in Fig. 2.9 and Table 2.3.2, at 95% Recall the deepCR-trained model has -13.19% and -7.59% loss in Precision on Gemini's 1×1 and 2×2 binning images, respectively, comparing to its performance on LCO images, while the Cosmic-CoNN model has consistent precisions of -1.67% and $+2.99\%$. It shows that the Cosmic-CoNN framework is superior in producing more generic models for unseen instruments not included in the training data.

Examples of detection discrepancy are shown in Fig. 2.11. The Cosmic-CoNN model is better at detecting complete CRs of arbitrary shapes, especially the "worm-shaped" CRs that frequently appear in the *GMOS-N/S* images.

The Cosmic-CoNN model's consistent performance on other CCD imagers also shows the large, diverse LCO CR dataset produces rich cosmic-ray feature coverage that could be effectively generalized to other ground-based instruments. Fig. 2.12 (top row) shows the robust detection result of a heavily CR-contaminated image from Gemini GMOS-N.

Bhavanam et al. [49] recently tested Cosmic-CoNN on DECam data [50] and showed it generalizes well to yet another unseen instrument - our Cosmic-CoNN model trained on the LCO CR dataset achieved a precision of 96.60% at 95.0% recall, a similar performance as in Gemini's 2×2 binning images (Table 2.3.2). Their improvement of adding attention gate modules [51] only brought marginal performance gain: 0.12% higher in true positive rate at 0.01% false positive rate and 0.07% higher in precision at 95.0% recall than training with the original Cosmic-CoNN framework. We argue potentially better performance from Cosmic-CoNN as [49] incorrectly trained on 256^2 pixel patches, which is against our training strategy discussed in Sec. 2.3.2.

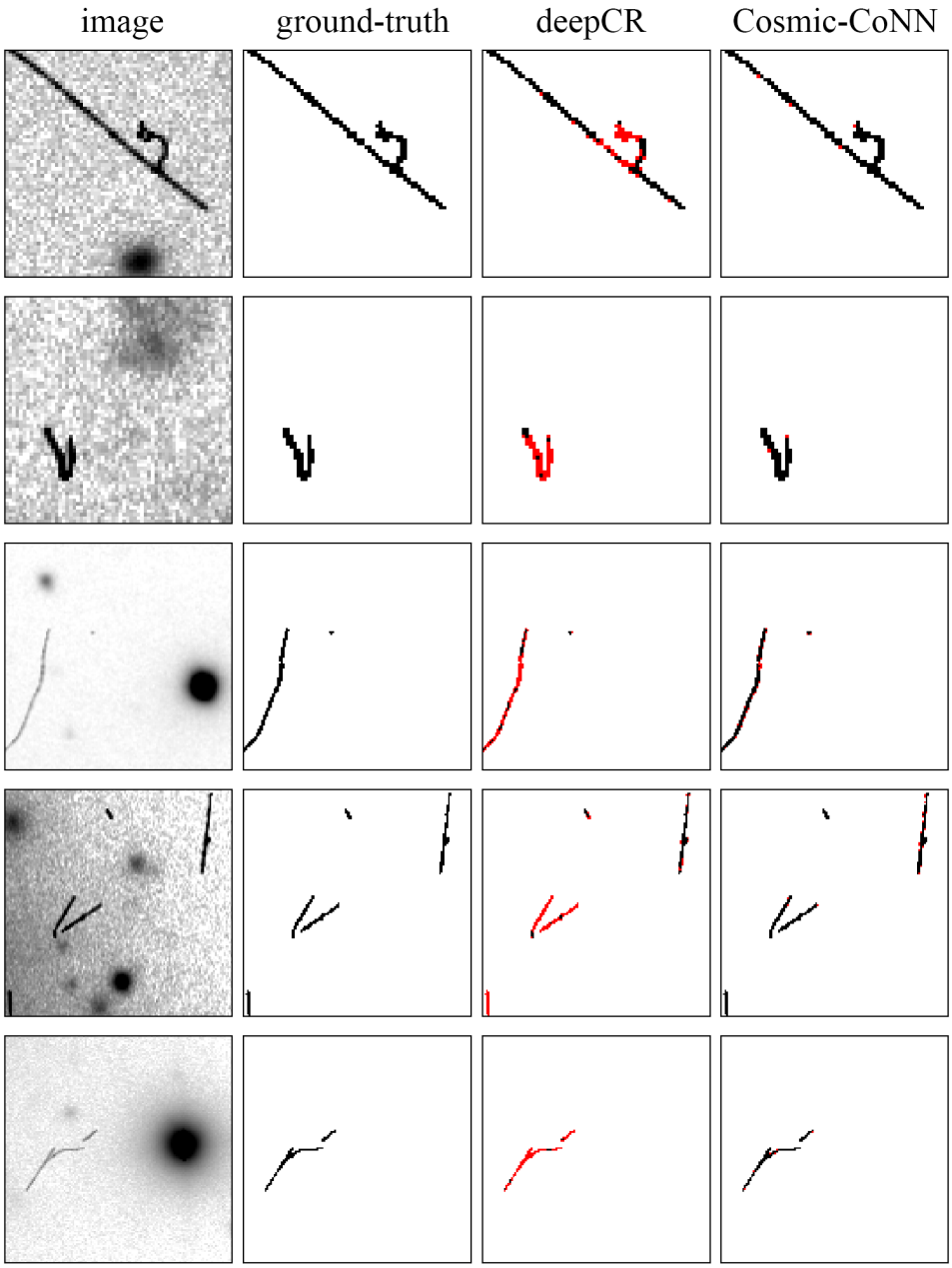


Figure 2.11: deepCR and Cosmic-CoNN were both trained on *LCO* data and tested on *GMOS-N/S* images that were never used for training. While they perform comparably on most CRs, we illustrate some examples that caused deepCR's performance loss on Gemini images (deepCR's 76.27% & 81.87% vs. Cosmic-CoNN's 92.03% & 97.69% in Table 2.3.2). Both models used the theoretical best threshold of 0.5 for binary masks. Incorrect or missing CR pixels are marked in red.

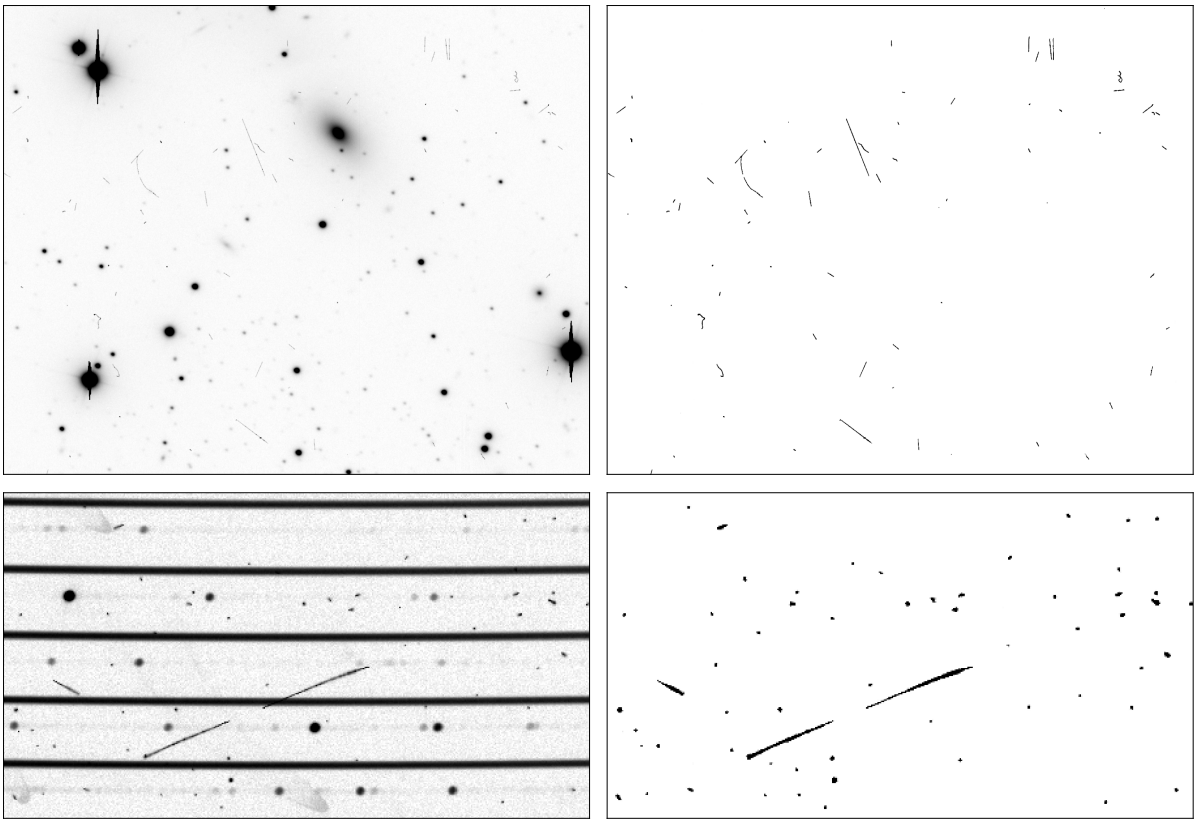


Figure 2.12: A pair of CR detection examples that shows both the Cosmic-CoNN model's generality and the framework's broad applicability. **(Top)** Cosmic-CoNN's generic ground-imaging model was trained entirely on LCO data yet all visible CRs in a new Gemini GMOS-N 1×1 binning image stamp are correctly detected regardless of their shapes or sizes. **(Bottom)** The Cosmic-CoNN framework also trains well on spectroscopic images and detects CRs over the spectrum robustly on a *LCO NRES* image. The horizontal bands in the left image are the spectroscopic orders, which are left out of the CR mask.

2.4.3 Space-based imaging data

We also trained Cosmic-CoNN on [29]'s *HST ACS/WFC* F606W dataset consisting of extragalactic field, globular cluster, and resolved galaxy observations to demonstrate the framework's broad applicability. The Cosmic-CoNN-trained model has better performance in all three types of observations comparing to the deepCR model (version 0.1.5), as shown in Table. 2.13. When testing model robustness on augmented images with random mirroring and rotation [42], we found more robust performance from Cosmic-CoNN

Data	Method	0.01% FPR	0.05% FPR
	(TPR loss w/ mirror+rotation)		
EF	deepCR	79.5 *(-0.2)*	88.6 *(-0.2)*
	Cosmic-CoNN	80.2 *(-0.1)*	89.0 *(-0.1)*
GC	deepCR	85.4 *(-0.6)*	93.4 *(-0.3)*
	Cosmic-CoNN	86.0 *(0.0)*	93.8 *(0.0)*
RG	deepCR	62.1 ***(-6.8)***	75.2 ***(-6.1)***
	Cosmic-CoNN	63.6 ***(0.0)***	76.3 ***(0.0)***

NOTE—All values are FPR (%) in fixed TPR. EF: extragalactic field, GC: globular cluster, RG: resolved galaxy

Figure 2.13: Table 3: We reproduced deepCR's results on *HST ACS/WFC* images to compare with Cosmic-CoNN. To test model robustness, we randomly rotated and mirrored the images and indicated each method's performance loss in *italic parentheses*.

with little or no performance loss, especially in resolved galaxy data (italic parentheses in Table. 2.13).

Unlike the LCO CR dataset which releases full-size images in `FITS` format, the F606W dataset sliced and stored images as 256^2 pixel stamps in `Numpy` arrays, so we were not able to test the effect of increased sampling stamp size on these data. [52] recently trained an all-filter *HST ACS/WFC* deepCR model on an extended dataset covering the entire spectral range of the ACS optical channel. Cosmic-CoNN supports loading deepCR models to use with our toolkit, instructions are available at `https://github.com/cy-xu/cosmic-conn`.

2.4.4 Ground-based spectroscopic data

Finally, we expand the Cosmic-CoNN framework to detecting CRs in single-exposure spectroscopic images, a task that has remained challenging for conventional methods. [53] was able to detect as many as 80% of the CRs in single-exposure, multi-fiber spectral images. Based on two-dimensional profile fitting of the spectral aperture, their method

takes about 20 minutes to process a $4K \times 4K$ pixel image. Cosmic-CoNN detects nearly all CRs in about 25 seconds on CPU and less than 5 seconds with GPU acceleration.

To prepare the data for deep-learning training, we modified our custom CR-labeling pipeline (Appendix 2.8.1) and produced a dataset of over $1,500$ images using repeated observations from the four instruments of *LCO*'s *Network of Robotic Echelle Spectrographs* (*NRES*) located around the world. We randomly sampled and reserved 20% of the data as the test set and used the rest for training and validation.

Cosmic-CoNN reaches 97.40% TPR at 0.01% FPR with a precision of 94.4% at 95% recall. Considering the high CR rates in spectroscopic images because of the 15 minutes or longer exposure time, the NRES model in fact demonstrates exceptional performance. A detection result example is shown in Fig. 2.12 (bottom row). We consider these results preliminary because the focus of this paper is on a generic ground-based imaging model and we will conduct thorough comparison with other methods in a future work. Nevertheless, the versatility of Cosmic-CoNN framework potentially paves a way for solving the CR detection problem in the accuracy-demanding spectroscopic data.

2.5 Toolkit

We have built a suite of tools to democratize deep-learning models in order to make automatic, robust, and rapid CR detection widely accessible to astronomers. The toolkit includes console commands for batch processing `FITS` files, a web-based app providing CR mask preview and editing capabilities, and Python APIs to integrate Cosmic-CoNN models into other data workflows.

The Python toolkit package is released on `PyPI`. We host the open-source Cosmic-CoNN framework on GitHub `https://github.com/cy-xu/cosmic-conn` with complete documentation including toolkit manual, developer instructions on using the LCO CR

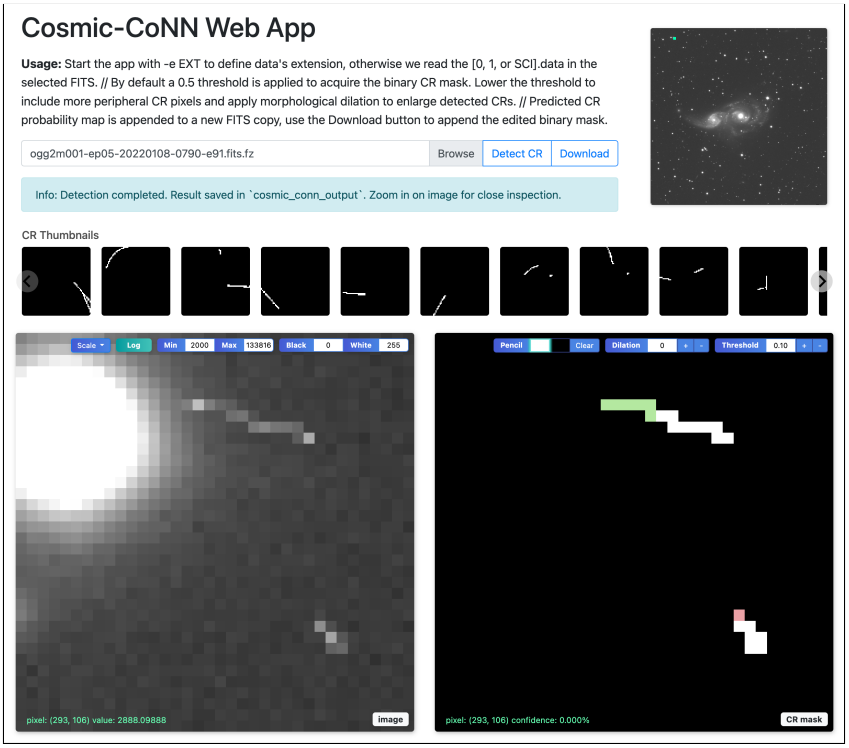


Figure 2.14: The interactive web app streamlines the workflow of CR detection, visualization, and mask editing into a single interface [54]. This tool is also useful to help users find the suitable threshold for new data. Users could adjust the threshold, apply morphological dilation, or perform pixel-level manual editing on the CR mask to acquire the desirable results for downstream analysis.

dataset and training new models. We also released the LCO CR dataset and the code used to generate the results to facilitate reproducibility.

Console commands are the most convenient way to perform batch CR detection on `FITS` files directly, e.g.,

```
$ cosmic-conn -i input -m ground_imaging
```

utilizes the generic `ground_imaging` model and the user can replace the argument with `NRES` or the path to a new model trained with Cosmic-CoNN for other types of data. The result is attached as a `FITS` extension. In terms of speed, Cosmic-CoNN provides more accurate prediction than conventional methods in comparable time on the CPU. Processing a $2K \times 2K$ pixels image takes $\sim$7.5s on a `AMD Ryzen 9 5900HS` laptop pro-

cessor. With GPU-acceleration, it takes only ∼0.8s on a high-end `Nvidia Tesla V100` GPU, and ∼1.2s on an entry-level `Nvidia GTX 1650` laptop GPU.

The `$ cosmic-conn -a` command starts an interactive CR detector in the browser, as shown in Fig. 2.14. We adopt the interface layout and controls from the `SAOImageDS9` [55]. In addition, we provide an array of CR thumbnails for quick navigation and the ability to edit CR masks in real time. The `JavaScript`-backed web app provides necessary tools for users to fine-tune the appropriate post-processing parameters for different instruments. The preview window supports various scaling methods like the `zscale` for better visualization.

Cosmic-CoNN is designed to be integrated in custom data pipelines. Let `image` be a two-dimensional `float32` array:

```
-----------------------------------------------
from cosmic_conn import init_model
# initialize the generic ground-imaging model
cr_model = init_model("ground_imaging")
# the model outputs a CR probability map
cr_prob = cr_model.detect_cr(image)
# acquire a Boolean mask with a 0.5 threshold
cr_mask = cr_prob > 0.5
-----------------------------------------------
```

Our `Python` APIs allows other facilities to integrate rapid CR detection into their data reduction pipeline. The framework checks if the host machine supports GPU-acceleration and prioritizes computation on GPU. Then it optimizes the detection strategy (full image or slice-and-stitch using smaller stamps) based on available memory without human intervention.

We are planning to deploy the web app on the cloud to provide GPU-accelerated CR detection as a free service. This will allow users to upload their failure cases to us to expand the training set and improve the model. In the current release, the web app is a local instance which does not collect or upload any user information.

2.6 Conclusion

In this work, we presented an end-to-end solution to help tackle the CR detection problem in astronomical images. The large, diverse LCO CR dataset produces rich feature coverage, allowing deep-learning models to achieve state-of-the-art CR detection on single-exposure images from Las Cumbres Observatory. The Cosmic-CoNN deep-learning framework trained generic CR detection models that maintain consistent performance on unseen instruments. Extensive evaluation showed the framework's broad applicability in ground- and space-based imaging data, as well as spectroscopic data. Finally, we released a toolkit to make the deep-learning CR detection easily accessible to astronomers.

Using the generic Cosmic-CoNN model as a pre-trained initialization, other facilities could fine-tune a model optimized for their own CCD imager with a lot less data. The LCO CR dataset also lays the foundation for a potential universal solution. By expanding our dataset with more instruments from other facilities, we are confident to see an universal CR detection model that achieves better performance on unseen ground-based instruments without further training.

The Cosmic-CoNN framework and the toolkit will be a valuable resource for the community to develop future deep-learning methods for source extraction, satellite detection, near-Earth objects detection, and more. These topics are not the focus of this paper but our improvements to the neural network made Cosmic-CoNN a suitable deep-learning architecture for these tasks, as we have seen in some preliminary experiments.

With the current Cosmic-CoNN model rejecting CRs that could be falsely recognized as astronomical sources, we could better profile the point spread functions in order to address the $\sim$1.21% excluded CR pixels in the next release of our dataset. We expect to see further improvement in the Cosmic-CoNN model.

As large surveys like the *Vera Rubin Observatory*'s *Legacy Survey of Space and Time*

(*LSST*) [56] go online, we will see an explosion of new data that requires automatic, robust, and rapid CR detection. With GPU-acceleration, deep-learning methods like Cosmic-CoNN will likely be the solution for future data reduction pipelines that is needed to process the over 100 terabytes of data produced each night from *LSST* and many follow-up facilities.

2.6.1 Facilities and Software Used

Facilities: *LCOGT*, *HST(ACS/WFC)*, *Gemini:Gillett*, *Gemini:South*

Software: `Astropy` [57, 58], `Astro-SCRAPPY` [47], `Cosmic-CoNN` [59, 60], `DRAGONS` [48], `reproject` [61], `Matplotlib` [62], `NumPy` [63], `scikit-image` [64], `SExtractor` [65], `PyTorch` [66]

2.7 Acknowledgments

We thank Yuxiang Wang, Jiaxiang Jiang, Chris Hellmuth, Keming Zhang, Jennifer Jacobs, and Tobias Höllerer for their discussion and feedback on this work. We thank Simon Conseil and the `DRAGONS` software [48] support team for their help in producing the *Gemini GMOS* evaluation dataset.

This work makes use of observations from the *Las Cumbres Observatory* global telescope network [67]. This works is also based on observations obtained at the international Gemini Observatory, a program of NSF's NOIRLab, which is managed by the Association of Universities for Research in Astronomy (AURA) under a cooperative agreement with the National Science Foundation on behalf of the Gemini Observatory partnership: the National Science Foundation (United States), National Research Council (Canada), Agencia Nacional de Investigación y Desarrollo (Chile), Ministerio de Ciencia, Tecnología e Innovación (Argentina), Ministério da Ciência, Tecnologia, Inovações e Comunicações

(Brazil), and Korea Astronomy and Space Science Institute (Republic of Korea).

Use was made of computational facilities purchased with funds from the National Science Foundation (OAC-1925717) and administered by the Center for Scientific Computing (CSC). The CSC is supported by the California NanoSystems Institute and the Materials Research Science and Engineering Center (MRSEC; NSF DMR 1720256) at UC Santa Barbara. This work was also partially funded by National Science Foundation grants IIS-1619376 and IIS-1911230.

2.8 Appendix

2.8.1 CR Labeling Pipeline

The ground-truth CR-labeling pipeline starts with searching for successive exposures of the same field. We acquire the publicly available scientific observations from *LCO*'s Science Archive[2] and filter the number of visits users requested (more than three but no more than twelve). It is unlikely a cosmic ray will hit the same pixel location twice, so every three consecutive exposures are saved as a sequence into a multi-extension `FITS` file for alignment and CR labeling, while maintaining all the header information for future community research. For higher signal-to-noise ratio and higher CR rates, we only used images with an exposure time of 100 seconds or longer. We further constrained the consecutive images to be taken within the same schedule molecule, the minimal *LCO* scheduler unit. Images from the same molecule ensure intervals between exposures are minutes or less, which minimize the variations in seeing conditions and point spread function (PSF). We reject a sequence whose background varies over $\sigma > 5$ between frames, as they are not stable enough to robustly identify cosmic rays.

[2]https://archive.lco.global/

We then reproject to align each frame in the sequence with `astropy/reproject` [68] using nearest-neighbor interpolation to ensure CRs are not distorted during re-sampling. Fig. 2.2 shows an image stamp from an aligned sequence. *LCO*'s `BANZAI` [33] data reduction pipeline have bias and dark frame subtracted to remove instrument signature, allowing us to use one CR-labeling pipeline across all *LCO* instruments. Let I be an image in the sequence then I's noise uncertainty σ_I is simplified to:

$$\sigma_I = \sqrt{|I| + N_R^2 + N_S} \tag{2.6}$$

where N_R is the CCD read noise, N_S is the sky background noise, which corrects for the background variation between exposures. We then approximate the median frame uncertainty Σ by performing median filtering at each pixel location across the uncertainties from the three frames I_1, I_2, and I_3 in order to reject the variance from the CR pixels:

$$\Sigma = \frac{Median(\sigma_{I_1}, \sigma_{I_2}\sigma_{I_3})}{\sqrt{3}}. \tag{2.7}$$

We update each frame I with sky subtraction $I := I - Median(I)$ before calculating the median frame M_I. We then define a deviation score that calculates how much each frame deviates from the median frame represented in Gaussian distribution:

$$\text{Deviation score} = \frac{I - M_I}{\sqrt{(\sigma_I)^2 + \Sigma^2}}. \tag{2.8}$$

Pixel locations with a deviation score > 5.0 are identified as bright CR pixels and labeled in a preliminary outlier mask. A morphological dilation of five pixels is applied to the outlier mask, and we use a lower threshold of > 2.5 to include the dimmer peripheral pixels around the CRs.

A key step to acquire the final CR mask is to remove false-positive outliers caused by

PSF wings and isolated hot pixels. We perform source extraction with `SEP` [69] on the CR-free median frame to acquire a robust source catalog. We then perform windowed background estimation to include the astrophysical source pixels in an ignore mask to reject false-positive outlier from PSF wings [70].

`BANZAI` provided a mask for permanent dead CCD pixels but we also noticed a very small fraction of remaining standalone hot pixels that are more likely to be Poisson noise or persistent pixels due to over saturation in previous exposures. Thus our last step is to reject isolated (single) hot pixel events to acquire the final CR mask. Different types of artifacts and rejected pixels, including 100 pixels ignored around CCD boundaries are coded and included in the ignore mask. Instruction on using the data pipeline, the LCO CR dataset, and the ignore mask coding rules can be found in the documentation `https://github.com/cy-xu/cosmic-conn`.

2.8.2 Ablation Study

An ablation study helps us understand how a building block or a design choice affects a machine learning system's overall performance. It applies or removes a single component in a controlled experiment while holding other parameters constant. We evaluate the proposed improvements discussed in Sec. 2.3 through variant models corresponding to Fig. 2.7 and present the quantitative results in Table. 2.15.

The complete ablation study (combining quantitative results from Table. 2.15 with training visualizations in Fig. 2.7) shows applying the proposed Median-Weighted loss function to the baseline method improves model performance on LCO data from 89.19% to 92.98%, at the same time improves training efficiency from 2980 to 2080 epochs, which validates that the new loss function does indeed provide a better model convergence path discussed in §2.3.1.

Method	Dice score > 0.85	LCO Precision	Gemini 1×1 Precision	Gemini 2×2 Precision
deepCR (baseline)	2980	89.19%	79.59%	84.88%
deepCR + Median-Weighted loss	2080	92.98%	78.76%	83.08%
deepCR + 1024^2px	n/a	89.35%	82.57%	86.55%
deepCR + GN	1420	90.82%	77.07%	89.30%
deepCR + 1024^2px + GN	1040	93.17%	84.54%	92.09%
Cosmic-CoNN (MW loss + 1024^2px + GN)	**380**	**93.40%**	**86.80%**	**94.37%**

Figure 2.15: Table 4: Cosmic-CoNN ablation study on *LCO* and *Gemini* imaging data. All variant models are evaluated with identical validation images and the same input stamp size. We gauge training efficiency by the number of `epochs` a model takes to reach a Dice score > 0.85 [37] during training, corresponding to convergence curves in Fig. 2.7. We discussed in Sec. 2.4 that Precision is less sensitive to the varying CR rates between different datasets than TPR at fixed FPR, thus we measure a model's Precision at 95% Recall on *LCO* and *Gemini* data to evaluate how well it generalizes to unseen data, corresponding to a model's performance at epoch 4000 shown in Fig. 2.7, higher is better.

While the Median-Weighted loss alone does not produce a more generic model, all variant models trained with the larger 1024^2 pixel sampling stamps demonstrated better model generality on the unseen Gemini data, especially the 1024^2px + group normalization (GN) combination that we discussed in §2.3.2. GN alone does not improve performance but mainly contributes to training efficiency, which is better visualized in Fig. 2.7 when compared with models that adopt the two-phase training.

The proposed Median-Weighted loss further provided the (1024^2px + GN) variant model a better convergence path to produce the Cosmic-CoNN model that excels in both training efficiency (from 2980 to 380 epochs) and performance on not only LCO instruments which were used for training (from 89.19% to 93.40%) but also Gemini instruments that were not included in training data (from 79.59% to 86.80% on 1×1 binning & from 84.88% 94.37% on 2×2 binning) among all variant models.

The ablation study shows each of our proposed improvements affects certain aspects of the machine learning system and their joint effect contributes to the generic and best-performing Cosmic-CoNN model suitable for the CR-detection task in ground-based astronomical data with variable conditions from multiple instruments.

2.8.3 Training Details

We implement the Cosmic-CoNN framework in `PyTorch` 1.6.0 [66] with Adam optimizer [71]. Models for the same type of observation are trained with identical data, random seed, and hardware. We use the `Nvidia Tesla v100 32GB` GPU for training. The large GPU memory allows us to maximize the batch size n in each iteration. All training settings are identical unless it is clearly specified for a variant model. Scripts to reproduce our experiments are included in the source code.

For LCO imaging data, we randomly sampled and withheld 20% of the training set for validation. An initial learning rate of 0.001 was used for all models. During training, we monitor the validation loss for each model and manually decay the learning rate by 0.1 when the loss plateaus. In the ablation study, we reduce the learning rate to 0.0001 at epoch 3,000 for all models. Models using group normalization adopt a fixed `group=8` for all feature layers. For the median-weighted loss we linearly scale the lower bound α from 0 to 1 over 100 epochs. We re-implemented deepCR with identical network and adopted the two-phase training that [29] used to train deepCR models. The Cosmic-CoNN batch normalization (BN) variant model also adopted the two-phase training. In order to make fair comparisons, all Cosmic-CoNN and deepCR models were carefully tuned, the best models were used for evaluation.

The Cosmic-CoNN model and variant models with 1024^2 pixels sampling stamp size used a batch size of $n = 10$ in the ablation study. deepCR and its variant models adopt 256^2 pixels stamp size with $n = 160$ to ensure the model sees the same amount of pixels in a mini-batch. For a dataset of N samples, models trained with batch size $n = 10$ updates $\lfloor \frac{N}{10} \rfloor$ times in an epoch but models trained with $n = 160$ only update $\lfloor \frac{N}{160} \rfloor$ times, which leads to unfair comparisons on training efficiency. We addressed this issue by sampling a subset of $\lfloor \frac{N}{16} \rfloor$ samples as an epoch for models with batch size $n = 10$.

For *HST ACS/WFC* imaging data, the Cosmic-CoNN model is trained on identical data as deepCR [29] but with a new `PyTorch` data loader that added random rotation and mirroring while sampling images. The larger GPU memory allowed us to use 256^2 pixels sampling stamp size with $n = 160$.

For *LCO NRES* spectroscopic data, the neural network is identical to the Cosmic-CoNN ground-imaging model. We used a stamp size of 1024^2 pixels with $n = 8$, an initial learning rate 0.0001, and manually monitor and decay the learning rate.

Chapter 3

Human-in-the-loop Image Processing

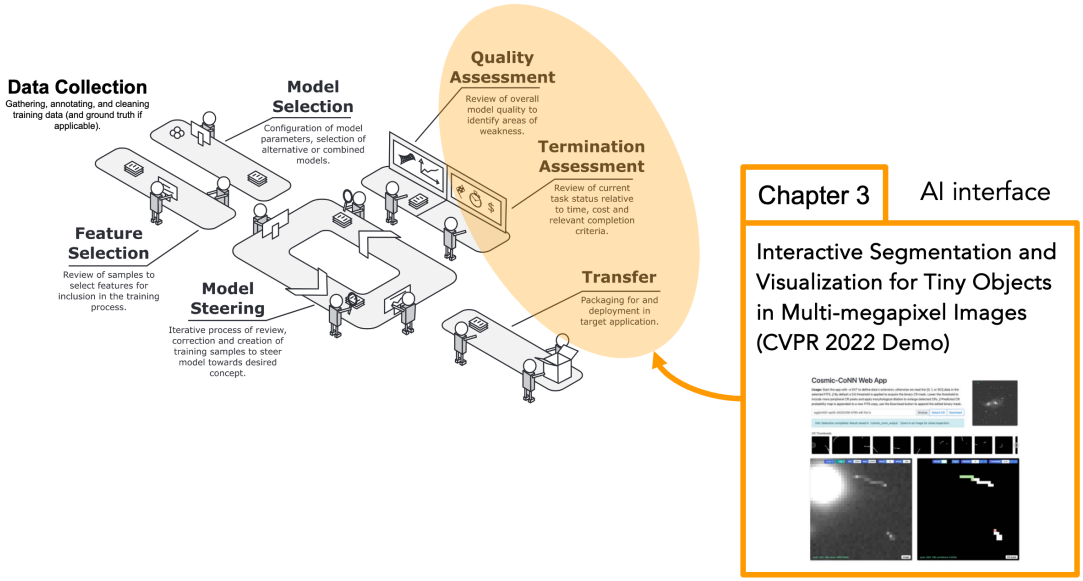


Figure 3.1: This chapter focuses on an AI-integrated interactive segmentation tool [54] that democratizes the computer vision detection capability and makes deep learning models accessible to scientists' image analysis workflow, improving user experience in real-world deployment. The thresholding visualization, pixel-level model confidence score, and direct output manipulation contribute to the quality assessment during training.

We present an interactive segmentation and visualization toolkit [54] that integrates deep learning model inference, HDR visualization, and segmentation mask inspection and editing within a single, user-friendly graphical user interface (GUI). The streamlined interface exemplifies how advanced computer vision technologies can be made accessible to domain experts without requiring deep technical knowledge in machine learning. For researchers who rely on the AI model's predictions for decision-making or downstream analysis, the AI-integrated interface provides additional information like the pixel-level confidence score and the ability to directly adjust AI's output with thresholding or manual editing. This human-in-the-loop collaboration enhances the black-box AI's utility while also allowing for human expertise to play a crucial role, ingesting confidence into the user's interpretation of the AI's behavior and fostering an effective partnership between human users and AI-assisted systems in the field of astronomical research and beyond.

Interactive Segmentation and Visualization for Tiny Objects in Multi-megapixel Images[1]

We introduce an interactive image segmentation and visualization framework for identifying, inspecting, and editing tiny objects (just a few pixels wide) in large multi-megapixel high-dynamic-range (HDR) images. Detecting cosmic rays (CRs) in astronomical observations is a cumbersome workflow that requires multiple tools, so we developed an interactive toolkit that unifies model inference, HDR image visualization, segmentation mask inspection and editing into a single graphical user interface. The feature set,

[1]The contents of this chapter have been previously published in **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. © 2022 IEEE. Reprinted, with permission, from C. Xu, C. McCully, B. Dong, D. A. Howell, P. Sen, and T. Höllerer, "Interactive Segmentation and Visualization for Tiny Objects in Multi-megapixel Images [54]," CVPR Demo Track, 2022, pp. 21447-21452

initially designed for astronomical data, makes this work a useful research-supporting tool for human-in-the-loop tiny-object segmentation in scientific areas like biomedicine, materials science, remote sensing, etc., as well as computer vision. Our interface features mouse-controlled, synchronized, dual-window visualization of the image and the segmentation mask, a critical feature for locating tiny objects in multi-megapixel images. The browser-based tool can be readily hosted on the web to provide multi-user access and GPU acceleration for any device. The toolkit can also be used as a high-precision annotation tool, or adapted as the frontend for an interactive machine learning framework. Our open-source dataset, CR detection model, and visualization toolkit are available at `https://github.com/cy-xu/cosmic-conn`.

3.1 Introduction

Semantic segmentation is not only a common computer vision task, but also a decades-old problem in astronomy. For astrophysicists whose research relies on observing the universe with optical telescopes and charge-coupled device (CCD) imagers, identifying cosmic rays (CRs) in their observations has been a challenging task[72, 73, 74, 47]. Telescope images can be a few megapixels or up to 3,200 megapixels [75], in contrast, CR-contaminated pixels are often just a few pixels wide. Because these bright pixels can be mistaken for real astronomical sources, it is necessary to reject them before further scientific interpretation of the data (see CR detection examples in Fig. 3.2 (5)).

Identifying tiny CRs in multi-megapixel images is only the first step. Astronomy telescope imagers are often cooled to operate below freezing temperature to minimize detector dark current and other noise sources [67], and these highly sensitive CCD sensors produce 16-bit floating point high dynamic range (HDR) images that require special software for visualization. Without a scientific visualization tool that supports native

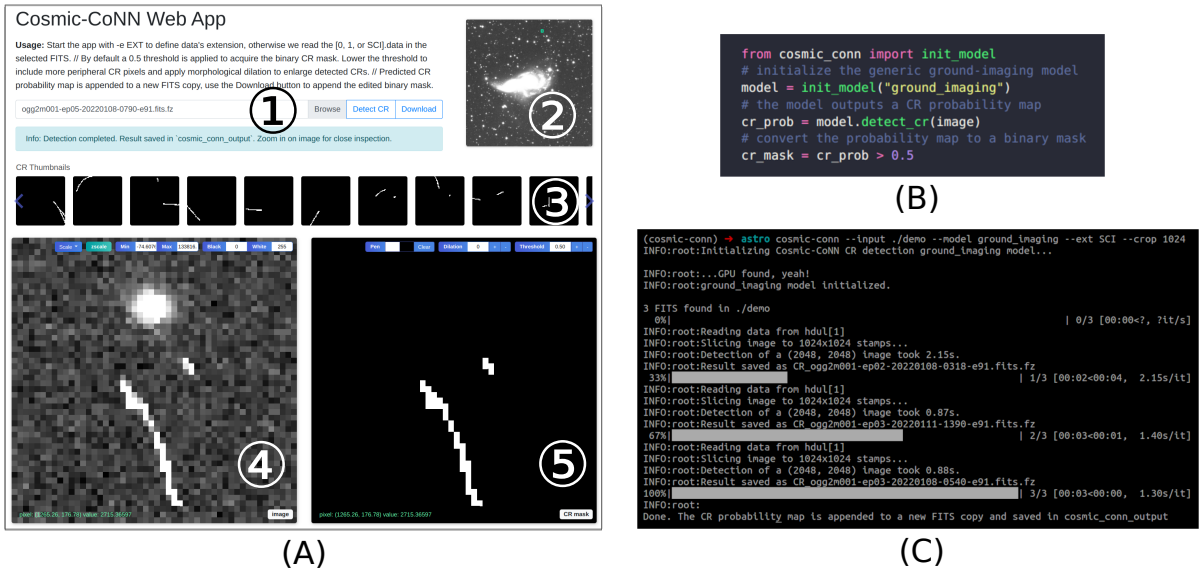


Figure 3.2: Our segmentation framework provides three user interfaces for different application scenarios: (A) the visualization toolkit has a graphical user interface (GUI) for model inference, interactive visualization, and mask editing, (B) clean Python library APIs for integration with user's data pipeline, and (C) a command-line interface for batch processing. GUI components: (1) File input and output; (2) Whole-image preview and navigation; (3) Thumbnails shortcuts to detected objects ranked from large to small; (4) Image window with various mapping (scale) algorithms and manual controls to visualize 16-bit floating point images; (5) Segmentation mask window with synchronized field of view with the image. The highlighted pixels are detected CRs. The user can adjust the visualization of the HDR data on the left while interactively editing the segmentation mask on the right.

integration with popular deep-learning frameworks, the detection and mask verification are divided into separate steps that involve exporting and reading files between different tools.

Given existing tools, the workflow of segmentation, image visualization, human inspection, and possible editing of the mask is a cumbersome process involving switching between multiple tools or software, making it worthwhile to develop a dedicated tool to streamline this workflow. In our video demonstration, we show an interactive process that involves continuous adjustments to both the science image and the segmentation mask to acquire the accurate coverage of a CR that might affect the analysis of an ad-

jacent stellar object. This level of seamless interaction was previously impossible if one were switching between different tools after each adjustment.

Computer vision researchers can integrate this visualization toolkit with other segmentation models to provide end users, especially domain experts who are not machine learning researchers, an interactive graphical user interface (GUI) (Fig. 3.2) in production. The streamlined workflow enables the user to do real-time segmentation, HDR image visualization, and interactive mask inspection and editing without switching tools. The GUI toolkit allows any user to benefit from deep-learning-powered tools without having to know programming. The browser-based tool can be readily hosted on a graphics processing unit-ready (GPU) server so users in the private/public network can enjoy GPU acceleration from any device (Section 3.3.2).

In addition, future tiny-object or high-precision segmentation tasks can adopt the interactive interface as an annotation tool for pixel-level labeling in multi-megapixel images, especially for HDR data. The Python backend allows native integration with popular deep-learning frameworks, with the potential to be an interface for Active Machine Learning and Interactive Machine learning (Sec. 3.5).

The discussion of detection algorithms is not the focus of this paper so we briefly introduce Cosmic-CoNN [76], our deep-learning segmentation framework deployed at Las Cumbres Observatory (LCO) for identifying cosmic rays in astronomical images. We curated a large, diverse dataset [77] of over 4,500 scientific observations from LCO's 23 globally distributed telescopes[2] [67]. In this dataset we discovered an extreme 1 to 10,000 class imbalance between CR and non-CR pixels that presented a challenge for previous machine-learning models. We proposed a novel loss function, and other improvements, to address this issue and increase model generalization. Our model achieved 99.91% true-

[2]LCO has 25 telescopes around the world now. Our research, started in 2020, used data from all 23 then-operational instruments. `https://lco.global/`

positive rate at a fixed false-positive rate of 0.01% on LCO instruments and maintains over 96.40% true-positive rate on data from another observatory, acquired with instruments that were never used for training (see [76] for details). Our CR detector has become part of LCO's BANZAI data reduction pipeline that processes hundreds of astronomical observations every day [78].

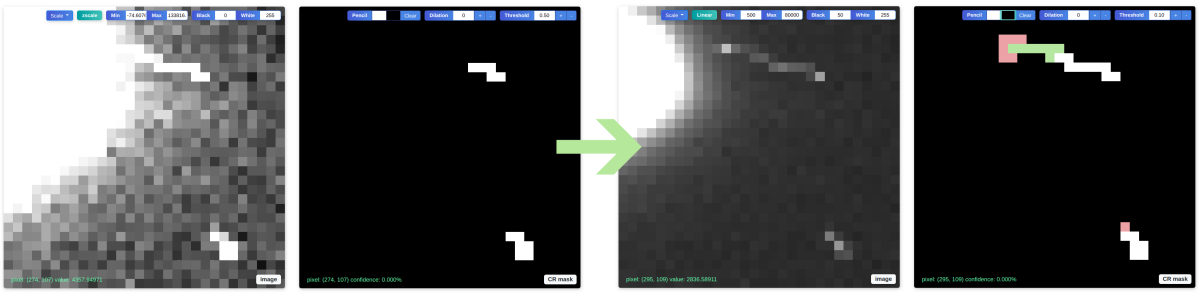


Figure 3.3: When the model or the default threshold does not produce ideal results (left), the user can adjust the HDR image mapping for better visualization, at the same time edit the mask interactively (right). The probability threshold and morphological dilation allow for global mask manipulation, while the pencil tool allows pixel-level mask editing. Pixels that are manually added/deleted by the user are marked in green or red, which will override the global manipulations.

Here we summarize the main contributions of our interactive visualization toolkit:

- We provide a streamlined workflow for CR detection, improving quality by enabling human-in-the-loop segmentation, and reducing the overall time cost of astronomical image analysis and interpretation.

- We address a use case that is common in scientific imaging, but not well-supported by existing tools: interactive segmentation in large, multi-megapixel HDR images with tiny objects.

- We release our software as an open-source package that can be deployed off-the-shelf with diverse image types and segmentation models, and can facilitate imaging research across many scientific disciplines.

In addition, we found the tool useful during the development of our tiny-object segmentation model. The interactive visualization provides timely feedback for changes to the image processing pipeline, making it a useful research-support tool in computer vision as well.

3.2 Usage

The visualization toolkit shown in Fig. 3.2 A is the key component to unify model inference, image visualization, segmentation mask inspection and editing into a single interface. It can visualize 2-dimensional NumPy arrays [63] and directly read FITS[3] files. It takes only 3 seconds to detect and render a 4-megapixel (2,000 by 2,000 pixels) 16-bit floating point image on a consumer laptop with a low-power NVIDIA RTX 3060 GPU.

The image window and segmentation mask window are always synchronized to an identical field of view (Fig. 3.2 (4) & (5)). This design provides a very useful reference for close inspection of tiny objects in large images. The user can navigate and zoom-in/out with mouse controls in any of the image windows, including the overview image (2). Thumbnail shortcuts (3) allow the user to quickly jump to and inspect detected objects, making it a unique design especially useful for locating tiny objects in very large images.

The image window (Fig. 3.3) provides multiple mapping algorithms to map (e.g., clip or normalize) 16-bit floating point data to 8-bit unsigned integers, including linear, logarithmic, and square-root scaling, as well as IRAF's zscale[4], an algorithm preferred by astronomers. The modular design of the image processing pipeline (Sec. 3.3.1) allows new mapping algorithms to be added easily. In addition, a user can manually assign

[3]FITS is an image and table format widely used for astronomical data `https://fits.gsfc.nasa.gov/fits_documentation.html`

[4]A reliable source for IRAF's zscale
https://docs.astropy.org/en/stable/api/astropy.visualization.ZScaleInterval.html

the minimum and maximum range to read from raw data for the versatility especially needed in HDR images. The bottom left corner of each window shows the mouse cursor's pixel-location value in original data and the predicted mask's confidence.

In the segmentation mask window (Fig. 3.3), a user can raise or lower the default 0.5 threshold to acquire a binary mask from the deep-learning model's predicted probability map $\in [0, 1]$. The user can then apply morphological operations like dilation to manipulate the mask globally, or use the pencil tool to manually edit the mask at pixel-level.

In the context of CR detection, the Download button will append the edited segmentation mask to the FITS file. This behavior can be changed based on the application. We can also change the communication mechanism with the deep-learning framework so the user can initiate the iterative labeling and training process in an active learning setting.

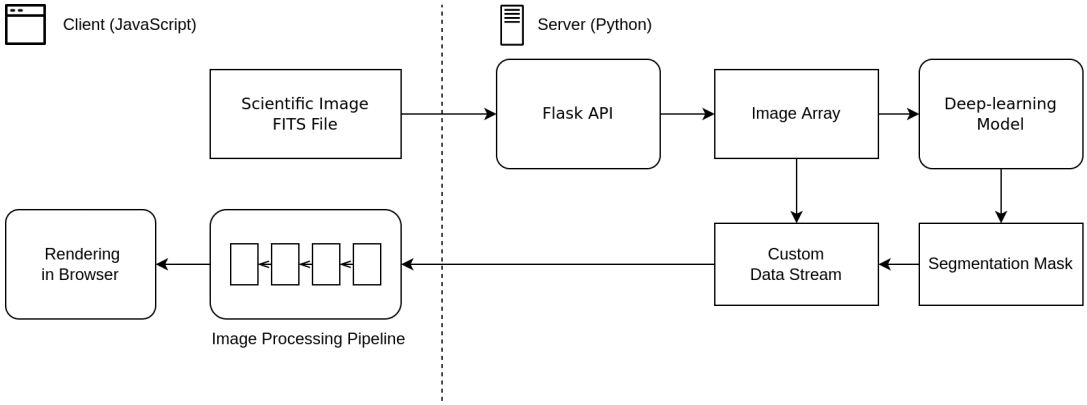


Figure 3.4: The interaction visualization toolkit's data flow architecture between the client and the server.

3.3 System Design

The visualization toolkit is powered by a Flask backend and JavaScript frontend (Fig. 3.4). The Python-based backend allows seamless integration with popular deep-

learning frameworks. We can run the server's instance locally or hosted on a cloud server for remote user access. The server-end only handles model inference and user instance management while the image processing pipeline happens entirely in the browser at the client-end. This design avoids overloading the server when hosted for multi-user access. The communication between the client and the server only happens at file uploading and downloading using a custom data steam to reduce the network delay.

3.3.1 Image Processing Pipeline

We adopt a modular design in the image processing pipeline to maximize the flexibility to add or remove image operations in the pipeline. The science image and the segmentation mask go through an ordered sequence of operations, and the modular design reduces computation and shortens the response time as the image is buffered after each operation – an adjustment in the middle of the pipeline will only trigger later stages to reprocess the image.

In the context of astronomical data, the pipeline will first apply user's manual min-max clipping to the raw data, then apply a three-sigma clipping to remove outliers (over saturation and dead pixels). By default the previously mentioned zscale algorithm is applied to map the 16-bit image to 8-bit before rendering in the browser.

The segmentation mask's pipeline is simpler as only one scalar threshold is applied to the probability mask to to acquire the binary mask. We use a separate mask to track the user's manual edits and combine with the binary mask before rendering in the browser.

3.3.2 Multi-user Support

Unlike many machine learning researchers, most of the real-world model end users do not have access to GPUs. With this in mind, we designed the GUI toolkit as a web-based

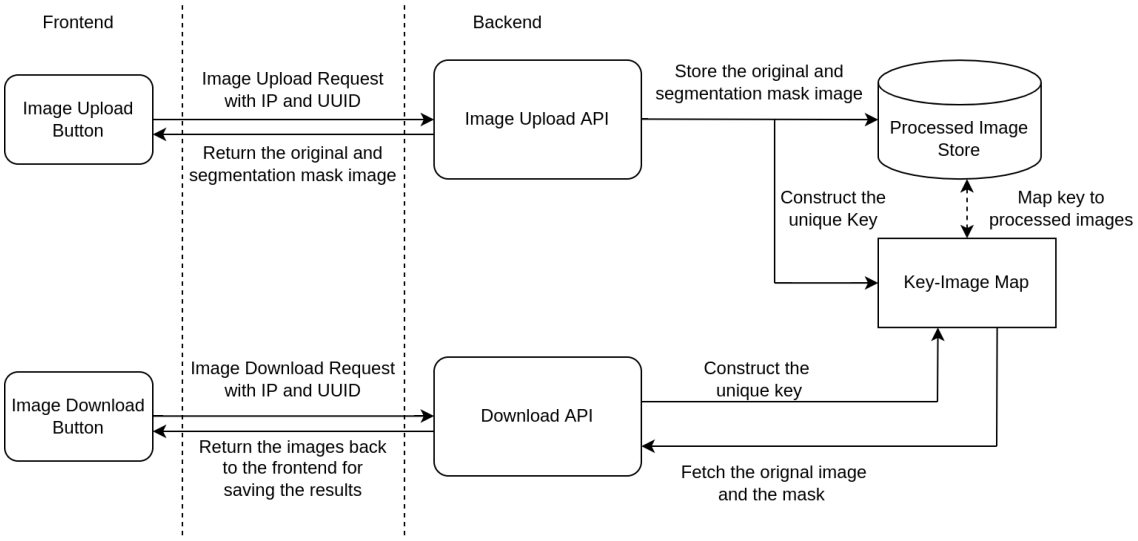


Figure 3.5: Frontend and backend architecture for multi-user interaction.

application to support GPU acceleration and multiple-user access from any device. In a large-scale deployment, additional cloud GPU resources could be recruited as necessary to support a higher number of users.

Fig. 3.5 illustrates our system architecture, which supports multi-user interaction via secure user sessions. In addition to user's IP address, a universally unique identifier (UUID) is sent to the server to identify each detection request, either through the upload request or the download request. A unique request key is constructed and the server will maintain a map of the key and temporary path of the uploaded files with the segmentation mask appended. When the user is done with editing the image and requests to download the combined results, the key is used to retrieve the correct file. In this way, we avoid race conditions when multiple users interact with the same deployed application.

3.4 Advantages Over Existing Tools

Our interactive segmentation and visualization toolkit has the following key features:

- A synchronized dual-window design (Fig. 3.2 A) and thumbnail-based image navigation (Fig. 3.2 (3)) enable inspecting and editing tiny objects in large images;
- Computer vision researchers can inspect the results interactively via the GUI toolkit to better understand the model's behavior and assist their research. They can also deploy a GUI segmentation tool for end users in production environment with little effort;
- The browser-based application can be hosted on the cloud or internal GPU server to support multi-user access and GPU acceleration from any device;
- The Python and Flask backend allow seamless integration with popular deep-learning frameworks. Researchers can adapt this GUI for high-precision annotation or Active/Interactive Machine Learning.

SAOImageDS9 [79] is a powerful FITS image visualization tool widely used in the astronomical community. DS9 inspired us to develop the GUI components in our toolkit. It supports various multi-frame layouts like tiling, blinking, and coordinates aligning. Despite active development, it remains primarily focused on visualization and we do not see an easy solution to integrate this standalone software with popular deep-learning frameworks.

ImageJ [80] is an image analysis program extensively used in the biological sciences. ImageJ2 [81] is a rewrite for multidimensional image data. It provides powerful image processing functionalities but requires a third-party plug-in to synchronize two image windows, and we haven't found a solution to make tiny object searching in large images as easy as the thumbnail shortcuts provided in our toolkit. ImageJ is versatile and general-purpose. Based on ImageJ, the later developed AstroImageJ [82] provides an astronomy specific image display environment and tools for astronomy specific image

calibration and data reduction. However, both tools were not optimized for the deep-learning segmentation workflow.

DeepImageJ [83] is a plugin to support the use of pre-trained deep-learning models in ImageJ. It provides access to various models in a biomedical model repository (BioImage Model Zoo), and allows basic deep-learning model inference. But it also carries over ImageJ's disadvantages we discussed above and is hard to integrate with popular deep-learning frameworks, especially for researchers who need interactive data analysis during the research stage.

ITK-SNA [84] is well known for 3D medical image segmentation, providing powerful functionalities from community contributions. But it lacks the support for deep-learning methods and the standalone software is hard to integrate with other frameworks.

3.5 Discussion

This demonstration highlights our three-in-one toolkit (segmentation, visualization, and editing) which streamlines the CR detection workflow and enables human-in-the-loop, interactive tiny-object segmentation in large, multi-megapixel, HDR images. In the future, we anticipate that user interfaces such as this one will be instrumental in the development of Interactive Machine Learning (IML) systems. Such systems are a promising approach for machine learning in domains where unlabeled data are abundant but annotations are expensive or difficult to obtain. The IML learning paradigm is especially beneficial in areas where domain knowledge is required, like biomedicine, astronomy, material science, etc., in which it is helpful for domain experts to steer the model training process. IML also reduces the overhead for scientists in various disciplines to train machine learning models [12]. Our interactive frontend and backend architecture is a step towards that direction.

Our dataset, CR detection model, and interactive visualization toolkit are open source and available at `https://github.com/cy-xu/cosmic-conn`. New features, such as instance segmentation and multi-file interface, are under consideration. We look forward to other computer vision researchers joining the open-source project to make this toolkit more useful for its various applications in astronomy, computer vision, interactive machine learning, and other research areas.

We appreciate the helpful discussion and feedback from Prof. Jennifer Jacobs, Jiaxiang Jiang, Alex Rich, Kuo-Chin Lien, and members from the Expressive Computation Lab of University of California, Santa Barbara.

Chapter 4

Forming Human-AI Teams for High-Stakes Tasks

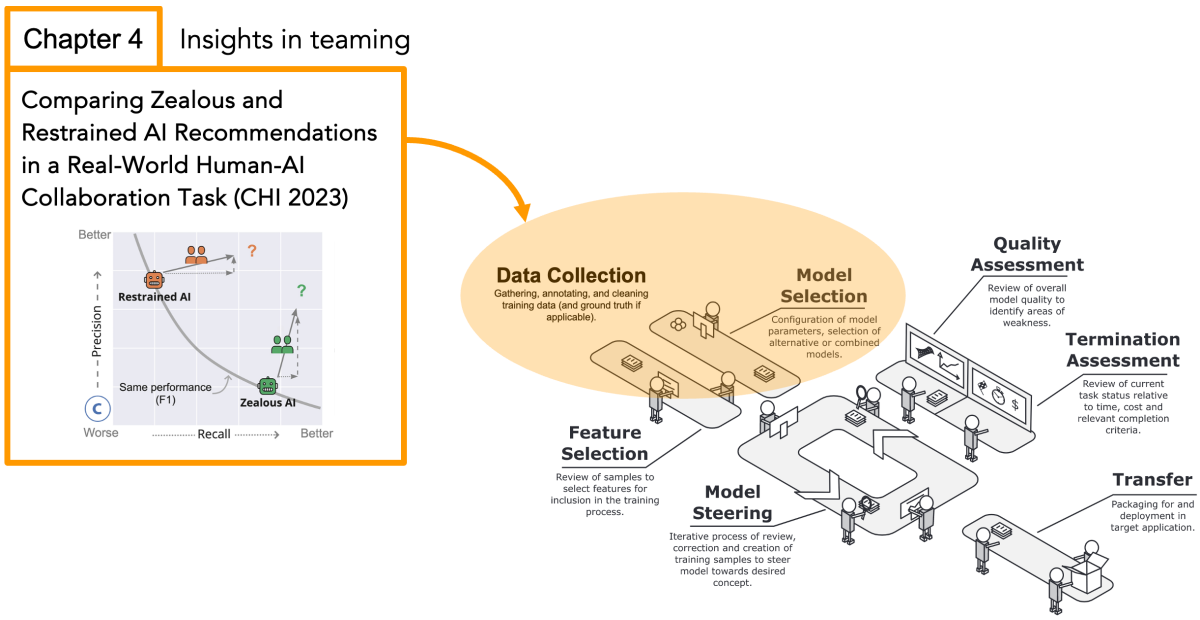


Figure 4.1: The two AI teammates, zealous and restrained AIs, discussed in this chapter are designed for a data collection task, specifically, video anonymization. The insights we learned from closely observing human annotators working and the user study provided guidance on model selection or optimization for human-AI teaming settings.

We introduce the concept of "Zealous and Restrained AI Recommendations" to harness the complementary strengths of human-AI collaboration and significantly enhance team performance. This research aims to generate new knowledge to assist AI researchers in selecting and designing machine learning models tailored for human-AI collaborative tasks, particularly in high-stakes scenarios such as video anonymization. Insights from a month-long study involving 78 full-time data annotators indicate that recommendations from off-the-shelf AI that is designed for autonomous workflows can adversely affect users' skills. This study provides a real-world exploration of how both novice and experienced users perform when paired with different types of black-box AI systems, examining the impacts of AI recommendations that prioritize either precision or recall on user performance. In addition, we propose a robust multi-object tracking algorithm that has been proven to be more suitable for human-AI teams in recall-demanding settings.

Comparing Zealous and Restrained AI Recommendations in a Real-World Human-AI Collaboration Task[1]

When designing an AI-assisted decision-making system, there is often a tradeoff between precision and recall in the AI's recommendations. We argue that careful exploitation of this tradeoff can harness the complementary strengths in the human-AI collaboration to significantly improve team performance. We investigate a real-world video anonymization task for which recall is paramount and more costly to improve. We analyze the performance of 78 professional annotators working with a) no AI assistance, b) a high-precision "restrained" AI, and c) a high-recall "zealous" AI in over 3,466

[1]The contents of this chapter have been previously published in **Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems**, 1–15. CHI '23. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3544548.3581282

person-hours of annotation work. In comparison, the zealous AI helps human teammates achieve shorter task completion time and higher recall. In a follow-up study, we remove AI assistance for everyone and find negative training effects on annotators trained with the restrained AI. These findings and our analysis point to important implications for the design of AI assistance in recall-demanding scenarios.

Figure 4.2: In video anonymization, face annotation and blurring is a high-stakes task that requires humans to check every frame. It demands high recall because one missed face can reveal a person's identity in the entire video. We can improve recall and reduce task completion time by forming a human-AI team. We may have two AIs with the same (F1) performance as shown in (c) but provide different sets of recommendations (a & b). A "zealous" AI would prioritize recall by suggesting more detections, even low-confidence ones. A "restrained" AI would only provide high-precision recommendations. Which AI teammate can help the human annotators finish in less time and with higher recall?

4.1 Introduction

Machine-learning-based artificial intelligence (AI) systems have exceeded human performance in certain applications. But in high-stakes domains where fully-autonomous AI is not at peak performance or not permitted, such as in clinical decision-making [85, 86, 3, 87, 88] or driver assistance [89, 90, 91], forming a human-AI team is a viable strategy to improve both efficiency and accuracy. AI can provide recommendations while human users maintain agency and control over the final decisions. Studies have shown the human-AI team is expected to achieve "complementary team performance" – the team

performance being better than either one alone [92, 93, 94]. But there are more questions than answers on which exact factors in the AI system affect the team performance and how.

Bansal et al. recently showed in simplified binary classification problems that **the most accurate AI is not necessarily the best teammate unless it helps to improve the team utility** [92]. But how about in more complex problems where the AI teammate is not simply better or worse for its accuracy? For example, in many computer vision problems, people determine the best-performing algorithms based on combination metrics such as the F1 score [95, 96], which can be broken down into two metrics – precision and recall [97, 98, 99]. Researchers can either balance the two metrics or prioritize one over the other to identify the best model for their application [100, 101]. Two AI systems can have the same F1 score but provide very different recommendations with different measures of recall (see a, b in Figure 4.2). The tradeoff between precision and recall puts them on different parts of the same F1 isoline (see Figure 4.2 c). Without additional context, one might argue that there is no better or worse between these two AIs.

In order to capitalize on complementary strengths of humans and AI when presented with tradeoffs in AI precision and recall, we need to be able to answer two questions: **1) for a given task, can we clearly identify if either high precision or high recall is more important than the other**, and **2) independent of importance, is it vastly easier or harder for humans to improve either precision or recall.**

Consider for example a pedestrian detection task in a driver assistance system: prioritizing the detection model towards either precision or recall will hurt the other. Human instinct tells us the risk of a missing detection could be lethal, so we should tune the AI system to prioritize recall, i.e., towards a "zealous" AI that provides more detections (recommendations), even the low-confidence ones, at the risk of more false positive er-

rors. In this context, the opposite "restrained" AI would only provide high-confidence detections and prioritize precision, but at the risk of more false negative errors.

In this work, we investigate how a high-recall zealous AI and a high-precision restrained AI can affect human-AI team performance in a real-world scenario. Compared to, say testing pedestrian detectors on the road, video anonymization is a similar but easier-to-test recall-demanding task. We set up a face annotation task for personally identifiable information (PII) protection that blurs human faces in a real-world video dataset [102]. PII protection is a critical task with increasing demand for both ethical research and abiding by regulatory requirements[2]. Similar to pedestrian detection, where the cost of a missing detection is very high, one unlabeled face in a single frame can reveal a person's identity in the entire video, if not the entire dataset.

This paper focuses on the common yet critical human-AI collaboration setting, in which recall is more important than precision. As for our second question, "is it vastly easier or harder for humans to improve either precision or recall?", an in-depth analysis of the video annotation workflow shows that improving recall is more costly than precision in this task since it is much harder for human annotators to draw a bounding box accurately than rejecting an incorrect one (see Section 4.3.2 & 4.3.3 for a full discussion).

The answers to our two questions for our task reveal **an optimization opportunity: the AI recommendation tradeoff between precision and recall can be used to exploit complementary strengths of the human and the AI in such collaborative tasks**. We posit that similar optimization opportunities exist for many other human-AI collaboration tasks. In addition, locating faces is a human instinct[3] that requires no specific training or domain expertise to get started, making face detec-

[2]E.g., The General Data Protection Regulation (EU) or The California Consumer Privacy Act of 2018 (CCPA)

[3]Here we refer to the ability to find human faces in a given image. We do not refer to recognizing people by face, which can be affected by Prosopagnosia (face blindness).

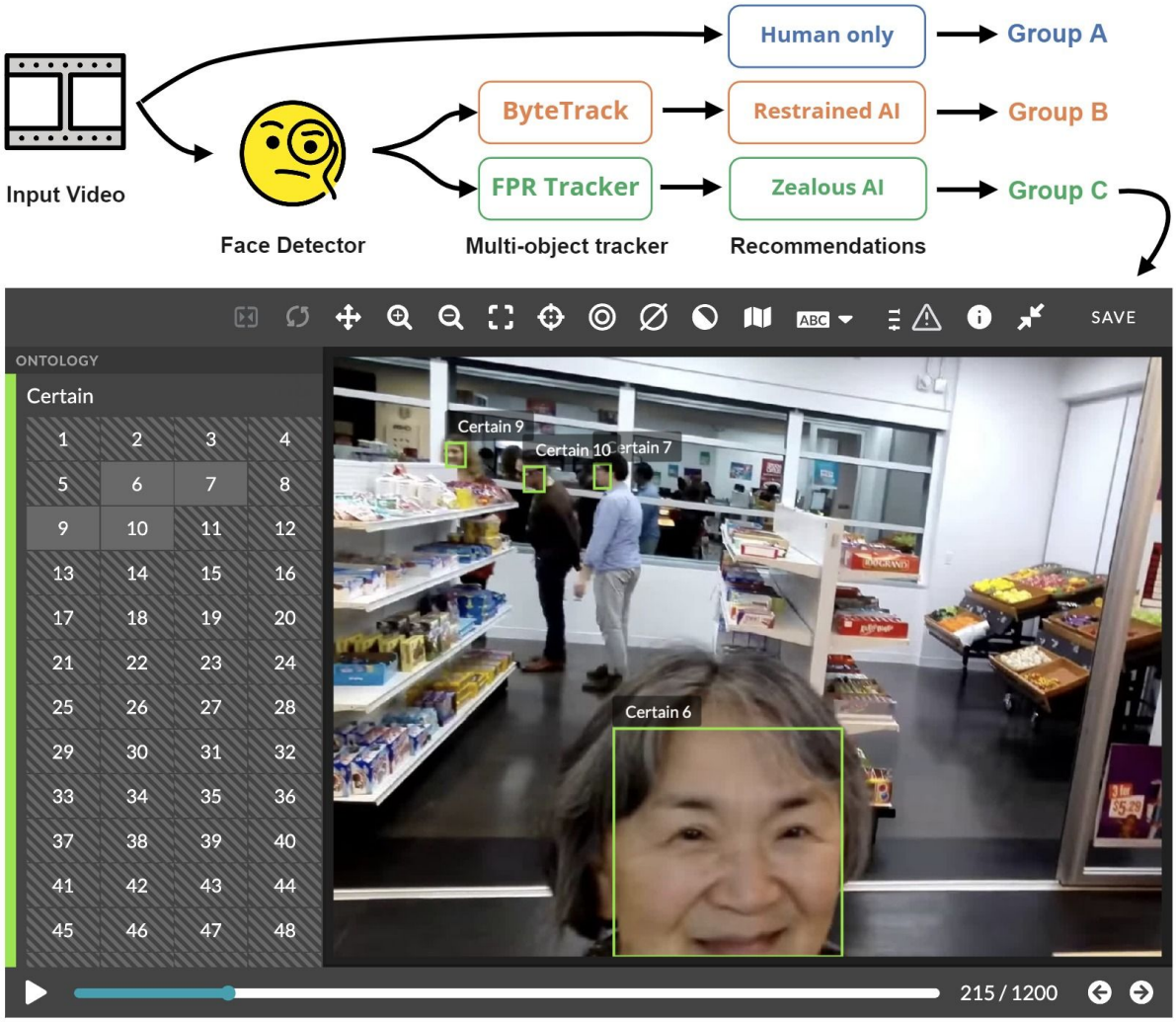


Figure 4.3: Data processing workflow for Part 1 of the study and the annotation tool user interface. The two AI teammates share the same face detector, which generates bounding box face detections for each frame independently. The ByteTrack tracker [103] and our proposed false-positive-robust (FPR) tracker define the restrained or zealous AI recommendations – they track the per-frame detections temporally to pre-annotate the videos as shown above. For the human-only workflow, annotators must manually draw a box and adjust its size and location across many frames.

tion a good candidate task to study the effects of different AI recommendations. The relatively small inter-personal differences also make the task a good representative of recall-demanding human-AI collaboration tasks.

Our large-scale empirical study had 78 professional data annotators spend over 3,466

person-hours[4] to submit a total of 2780 annotated 30-second videos. The between-subjects study split the annotators into three 26-people treatment groups. Detailed worker profiles ensured similar average experience between the groups (details in Section 4.4.2). Each participant annotated human faces in 36 real-world videos of a variety of activities (see examples in Figure 4.5). We measure each group's annotation quality and task completion time. Any improvement in time is very meaningful for annotation tasks not only because of the cost. Fatigue induced by long working hours may also cause a decline in quality.

In Part 1 of the **two-part study**, the three groups of annotators processed the same 24 videos, each with a) no AI assistance, b) pre-annotated bounding boxes recommended by the restrained AI, or c) the zealous AI. Figure 4.3 summarizes the treatment groups and shows the annotation tool's interface. In Part 2, the three groups annotated another 12 videos but all without the AI's help. This design **allows us to learn how prior human-AI collaboration experience can affect user skills**, should they lose access to AI recommendations in the future. The two-part experiment aims to answer the following research questions:

Q1 Can the human-AI teams achieve "complementary team performance" in this task?

Q2 Which AI helps annotators be more efficient, i.e. save time?

Q3 Which AI helps annotators achieve higher recall?

Q4 Will collaborating with an AI improve or hurt user skills?

We will answer each of the research questions in Section 4.5. Here we summarize this work's contributions:

[4]Our system logged 3,466 person-hours of annotation work, which does not include pilot studies, training sessions, and answering multiple questionnaires. On average it took the 78 annotators three to four weeks to finish the entire study.

- We propose the concept of restrained and zealous AI recommendations to compare the tradeoff between precision and recall in tuning AI-assisted decision-making systems and investigate how they affect human-AI team performance in high-stakes recall-demanding tasks.
- We design a large empirical study to compare the restrained and zealous AI on a face annotation task for video anonymization with 78 professional data annotators. The two-part experiment yielded significant findings to inform future AI assistance design for recall-demanding tasks.
- The analysis of 3,466 person-hours of annotation work reveals significant findings:
  - Our study serves as a real-world case study of complementary team performance (cf. [9, 6, 2, 7]).
  - Identifying the complementary strengths of both human and AI teammates for a task is key to better team performance. The recall-demanding task and the higher cost of improving recall motivated us to propose the zealous AI, which provides high-recall recommendations and leads to significantly better task completion time and recall.
  - The follow-up study demonstrates that naively pairing humans with an AI system designed for autonomous settings without optimizing it for the task at hand or for the human-AI workflow could potentially have a negative training effect on the users.

4.2 Related work

Factors affecting human-AI team performance. While human-AI teams have been studied extensively from various perspectives like in crowdsourcing settings [6, 5],

computer vision tasks [6, 3, 2], high-stakes tasks [2, 104, 105, 106], and real-world tasks [6, 9, 2, 107, 108, 7, 88], we still have more questions than answers on exactly which factors affect team performance and how. Researchers have looked into factors like users' mental models [109, 104], user expectations [110, 111, 112], cognitive biases [113], model updates during collaboration [105], model accuracy [111, 92], model interpretability or explanations [114, 115, 116, 8, 117, 118, 93], as well as the tradeoff between accuracy and interpretability [86]. Studying user's trust and appropriate or inappropriate reliance on AI [119, 85, 120, 121, 122, 106] is another important direction.

This paper is aligned with works that focused on the tradeoff between precision and recall in AI recommendations and its effect on team performance. Kay et al. [110] introduced the acceptability of accuracy as a new measure and survey instrument to connect classifier evaluation to users' subjective perception of accuracy. Kocielnik et al. [123] compared two 50%-accurate AI-powered scheduling assistants – one avoids false positive errors, and one avoids false negative. This is a similar design as for our restrained and zealous AIs – their study found that false positive errors are more acceptable by participants, which corroborates the overall better performance we observed in the zealous AI group, who also dealt with more false positive errors.

Balancing precision and recall to compare two real-world AI systems in a human-AI collaboration task is not easy, previous works derived insight from hypothetical systems or manually balanced recommendations [110, 123]. In this work, we provide a real-world user study by observing how 78 professional users would interact with two high-performance face tracking AI systems that are tuned to truthfully portray the realistic tradeoff between high-precision and high-recall on a recent egocentric video dataset.

Face detection. The annotation platform we used has a built-in face detector, RetinaFace [124], integrated for autonomous workflows. Our literature search found RetinaFace remains a top-ranking method on the WIDER FACE benchmark [125]. Because

more recent methods do not provide significant performance improvement, we continue to use RetinaFace as a consistent baseline to compare with our algorithmic improvements in tracking.

Multi-object tracking. In the AI-assisted face annotation task, the AI teammate provides annotation recommendations for users to review. Conventionally a face detector provides per-frame face bounding boxes and a multi-object tracking (MOT) algorithm produces continuous tracks of the same object across frames. This is known as tracking-by-detection. Recent MOT methods like TransTrack [126], DETR [127], Deformable DETR [128], TrackFormer [129], and TransMOT [130] etc. all move toward the end-to-end Transformer-based [131] architecture. However, these black-box MOTs share the same drawback as they are designed for fully-autonomous settings. Similar to Caruana et al.'s observation that modular system provides better transparency [86], the two-part tracking-by-detection frameworks actually provide us the interpretability and flexibility to steer the output recommendations as needed, so we can produce restrained and zealous AI recommendations for comparison. We reviewed state-of-the-art methods in related multi-object tracking benchmarks [132, 133, 134] in search of a multi-object tracker suitable for a human-in-the-loop annotation workflow. ByteTrack [103] is a conventional tracker that outperforms numerous Transformer-based trackers mentioned earlier.

Video annotation. While there are various public video annotation platforms or tools to choose from [135, 136, 137, 138], we use a proprietary video annotation tool to gain access to professional data annotators who are already familiar with the specific tool from their past project experience. This tool has Linear Interpolation [136] activated by default, which provides semi-automatic assistance by linearly interpolating a box between two manually annotated key frames. In this study, all participants, including annotators who review AI's annotation recommendations have access to this functionality. Linear Interpolation is also an ideal baseline as all participants have sufficient experience using

it. We will refer to this basic setup as human only, the baseline method, or the manual method in the rest of the paper.

4.3 Algorithm choices and pilot studies

4.3.1 Precision and recall in multi-object tracking

Precision, recall, and F1 are important performance metrics that can describe the characteristics of a model and are central concepts in this work and other human-AI research [110, 123]. Specifically, in the context of annotating and tracking faces with bounding boxes in videos:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{\text{Face boxes correctly drawn}}{\text{All boxes drawn by the user (or the AI)}} \tag{4.1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{Face boxes correctly drawn}}{\text{All ground truth face boxes}} \tag{4.2}$$

$$\text{F1} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{4.3}$$

where the TPs are true positives, face boxes that were correctly drawn. The FPs are false positives, boxes drawn by the AI or user which did not match real faces properly. The FNs are false negatives, where there is a real face, but the box is missing.

The F1 score is the harmonic mean of the precision and recall (Equation 4.3). We visually introduced the concept of this function using three methods that have the same F1 score in Figure 4.2 (c). Put simply, a video pre-annotated by a high-recall method (zealous AI) would have more false-positive boxes – the user will make more rejections but add fewer missing boxes. A video pre-annotated by the high-precision method (restrained AI) would provide mostly correct boxes but the user will need to add more missing boxes.

We are interested in how users will perform differently given restrained or zealous AI recommendations in an AI-assisted face annotation task. While it is easy to generate

high-precision annotations by simply avoiding low-confidence detections, it is hard for trackers to produce high-recall results while maintaining a similarly high F1 score at the same time. This motivates us to propose a tracking algorithm that pushes recall to the limit, but aims to maintain a similar level of F1 score. We take advantage of the fact that **our tracking results will be reviewed by human annotators, allowing us to make targeted optimizations**. We test our ideas of a user-friendly tracker with professional annotators through pilot studies. Observing how users work with trackers allows us to further improve the algorithm.

4.3.2 Pilot studies

We conducted two pilot studies to observe how professional data annotators work with AI recommendations. Annotators were tasked to draw bounding boxes around potentially moving or blurred faces of any size in a 1,200-frame video sequence of a busy shopping scene in both sessions (similar to hard videos in the formal study). We provided training material on how to review recommendations from the AI for the face annotation job. The annotation tool user interface is shown in Figure 4.3. With their consent, we recorded their screens to keep track of mouse movements and other user habits. Each session included ten different users with above-average experience. Both pilot studies concluded with a survey about experiment design and their experience. The two pilot sessions were spaced two weeks apart to test algorithm and design improvements.

Users' screen recordings helped us observe the following user habits and behaviors that are not possible to be identified solely from the results:

- *Certain bad recommendations cost most of the human review efforts.* Following the Pareto Principle [139], annotators in fact spent most of their time and effort amending a small fraction of AI recommendations. The tiny bounding boxes (see

examples of three tiny faces in Figure 4.2), duplicate detections (often clustered), and temporally sparse detections (short tracks) are the most costly recommendations. Addressing these issues allows annotators to have better continuity in their workflow.

- *Model explanation should not increase task complexity.* Initially, we offered model explanations using "Certain" and "Uncertain" labels based on the face detector's confidence, hoping this can assist users' decision-making. But video recordings and user feedback revealed that the extra information in fact increased the task complexity and caused unnecessary confusion. This design was eventually not considered in the formal experiment.

Observing how human annotators review AI recommendations (bounding box pre-annotations) in multi-object detection and tracking tasks inspired us to **break the complex workflow into three fundamental user actions: *accept***, ***reject***, or ***solve***, each coming with a higher cost in time. Figure 4.4 explains each action's time complexity. We can connect these three actions with our two main objectives (time and recall) to make **a simple deduction to identify the human-AI complementary strengths** in this task:

1 ***reject*** improves precision and ***solve*** improves recall. A correct ***accept*** improves both.

2 It takes the AI constant time to ***solve*** additional cases (give more recommendations) with a downside of more false-positive boxes for humans to reject.

3 Humans are faster at ***reject***ing a false-positive (incorrect) box than to ***solve*** a false-negative (missing) box.

4 We also know recall is more important than precision in video anonymization tasks.

Figure 4.4: When reviewing the AI teammate's recommendations (green bounding boxes), a user takes one of the three actions for each box: ***accept***, ***reject***, or ***solve***. In video annotation, because **the boxes are temporally tracked across many frames, each action's time complexity is drastically different**, note the two types of *Solve* in frame 0 can come at different cost, too.

ID:1 – A user can *accept* the true-positive track ID:1 boxes without any action.

ID:2 – The entire false-positive ID:2 track can be rejected with two mouse clicks by deleting the ID in any of the frames, which is $O(1)$ in time complexity.

ID:3 – False-positive recommendations, like track **ID:3**, are the most time-consuming to ***solve***: the user can delete and redo this face, or manually adjust every frame until the AI's pre-annotation becomes acceptable with $\geq n$ mouse clicks, $O(n)$ where n is the number of frames.

ID:4 – In frame 0, to *solve* the false-negative missing box for the left-most person, a user needs to manually draw a box and adjust its location and size until the AI-suggested box **ID:4** comes in with $\leq n$ mouse clicks, $O(n)$.

5 Thus, **a clear path to better human-AI team performance is to delegate more *solve* actions to the AI, so the human's overall effort is reduced by doing more easy *rejecting* and only *solving* the most challenging faces**.

4.3.3 The false-positive-robust (FPR) tracker

We adopted a tracking-by-detection system to produce face pre-annotations (Section 4.2), the two-part system design allows us to feed the same per-frame face detection from RetinaFace [124] to different downstream multi-object trackers like the ByteTrack [103] or our own designs for a fair comparison. Learning from our pilot studies ob-

servations, we propose the false-positive-robust (FPR) tracker that specifically provides user-friendly annotation recommendations. We use **the following unconventional strategies** to design the FPR tracker that can take overwhelmingly noisy detections with a high false positive rate as input but outputs "clean" tracks for a human-in-the-loop workflow:

- To improve the AI's recall, we apply an **extremely low threshold ($t \geq 0.01,\ t \in [0, 1]$) on the face detector's confidence score** to keep any potentially useful detected boxes. This is not a viable solution for Autonomous AI systems but we are working in conjunction with a human.

- The consequence of such a low face detector threshold is **clusters of overlapping boxes** on small faces. Our solution: for each cluster, we perform non-maximum suppression [140] by only keeping the single bounding box with the highest confidence score because in most cases they are duplicate detections on one true face. This step also improves the AI recommendations' precision.

- Finally, based on our observation that **the majority of temporally sparse detections are false positives** induced by the low threshold, we remove any tracks that are shorter than m consecutive frames so they do not interrupt users' continuity. We used $m = 10$ in the FPR tracker. Although some true-positive faces are also removed, users are much faster at solving an unlabeled face from scratch than filling the gaps between temporally sparse detections.

To design the experiment, we also need a restrained AI that generates recommendations of similar performance (F1 score) but with high precision. This is done by using only the high-confidence ($t \geq 0.8, t \in [0, 1]$) face detections with ByteTrack. To ensure fair comparison and reduce moving parts in our systems, we use the same face detection

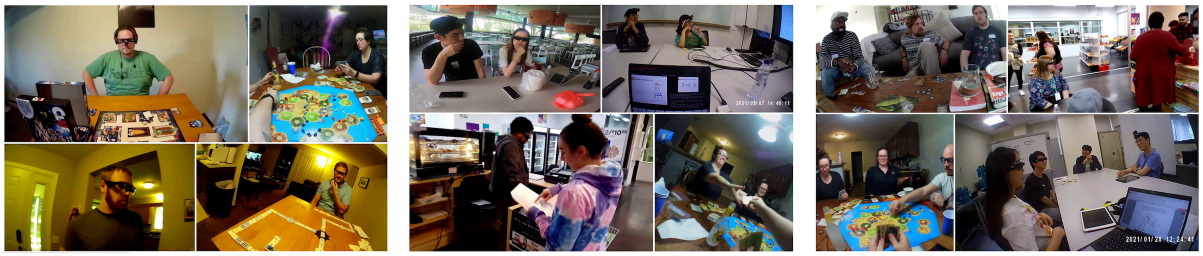


Figure 4.5: Screenshot examples of Ego4D videos [102] used in our face annotation experiment. Easy videos include about one face to annotate in each non-empty frame. Medium videos include about two faces. Hard videos include three or more faces. Videos with more faces are expected to take longer time to finish. The study results show shorter to longer completion times for Easy, Medium, and Hard videos in both parts (see Figure 4.7 and Figure 4.8), demonstrating that our video difficulty categorization is reasonable and performed as expected. We also considered scene diversity, box size (smaller faces are harder), and camera movement intensity (more movement is harder) to ensure a balanced difficulty distribution in selecting the specific videos.

model RetinaFace [124] for both AI teammates. It is the two different (fully transparent) trackers we apply that push the AI recommendations towards either high-precision or high-recall (Figure 4.2).

Note that we were only able to optimize the FPR tracker and ByteTrack through pilot studies because the ground truth data was not available for the 36 videos used in the user study. After the study, we aggregated the annotations from all 78 participants (2,780 submissions in total) to form an expert-reviewed consensus to serve as the ground truth. It turns out the zealous AI recommendations (FPR tracker) yielded an F1 score of 90.9% and the restrained AI (ByteTrack) had an F1 score of 93.4%. While the two AIs did not provide identical initial performance for their human teammates, we achieved the goal of two distinctive high-recall and high-precision AIs (Figure 4.6). The performance gap also provided us additional evidence to support our previous deduction on the zealous AI being the superior choice for this task, which we will discuss in Section 4.5.1.

4.4 Experiments

In this work, we aim to investigate how restrained and zealous AI recommendations will affect human-AI team performance. We are also curious if the collaboration experience with an AI teammate can affect users' skills, should they lose access to AI assistance in the future. We design a two-part empirical study to test the restrained and zealous AIs in a recall-demanding high-stakes task.

4.4.1 The task and data.

Face annotation for video anonymization is a perfect example of recall-demanding tasks – a missing face in a single frame can reveal a person's identity in the entire video. The high-stakes nature requires humans to annotate or verify every frame, yet the manual process will become the throughput bottleneck. The tedious process and long hours may also fatigue annotators and cause a decline in quality. In addition, **because the task of locating faces requires no specific training or domain expertise, it should help the generalizability of our observations** to other AI-assisted annotation tasks or even to other recall-demanding human-AI collaboration tasks.

In our human-AI collaboration setting, the AI teammate provides recommendations in the form of bounding boxes (see examples in Figure 4.3), and a user reviews each of the AI's pre-annotations to make one of the three decisions shown in Figure 4.4. We evaluate users' performance on the two most important metrics for face anonymization: **task completion time** and **recall**.

To test different AI recommendations in a real-world setting, we curate 36 first-person videos from a large-scale egocentric video dataset Ego4D [102]. Privacy has always been a major concern for datasets collecting human activities so first-person videos are ideal for this study. The videos we selected include various indoor social activities that are

suitable for benchmarking face detection and annotation tasks. Each video clip is 30 seconds long, or 900 frames. We estimate each video takes about 30 minutes to one hour to fully annotate, depending on its difficulty.

The different annotation methods (without or with different AI recommendations) adopted by the three treatment groups are the first level of independent variables that we will discuss in the next section. The second level of independent variables that can affect users' performance is the difficulty of the videos. We divide the videos into Easy, Medium, and Hard categories based on the average number of people one needs to track simultaneously in non-empty frames (see examples in Figure 4.5). We also considered factors like scene diversity, bounding box size, and camera movement intensity that affect the annotation difficulty in a more subtle way. Based on this overall difficulty ranking distribution, we ensure Part 1 and Part 2 videos are not only similar in content but also consistent in annotation difficulty.

We generate the bounding box ground truth by aggregating the crowd's annotations to reach a consensus, which is further reviewed and refined by a domain expert. We used an equal number of manual and AI-assisted submissions for each video to generate an unbiased ground truth.

On task completion time, **annotators are advised to finish each video without taking breaks longer than five minutes** but we still need to reject outlier video completion times caused by a known limitation of the annotation tool – the timer continues if an ongoing task window was left idle, or the timer will reset if the annotator continues from previously saved progress. We adopted median absolute deviation (MAD) [141] by comparing each video's completion time within each group to reject 420 out of 2780 (15.11%) completed videos, including completion times that are less than six minutes (the minimum time needed to verify each frame) or longer than median + 3 * MAD. The rejected videos also include all 36 submissions from one particular problematic user, see

Group	Novice	Veteran	Part 1 method	Submissions	Part 2	Submissions
A	11	14	Human only	602	Human only	299
B	14	12	Restrained AI + Human	619	Human only	304
C	13	13	Zealous AI + Human	621	Human only	299

Table 4.1: In the two-part study, the three treatment groups use different methods in Part 1, but we remove all AI assistance in Part 2. The novice and veteran workers represent a balance of different user expertise in each group. The submission numbers are the 30-second annotated videos each group finished. Note that Group A is one user short as a particular worker was later rejected because of repeated bad submissions.

Section 4.6.2.

4.4.2 Participants and three treatment groups.

A total of 78 in-house professional data annotators completed our study. It is important to note that in this project **they are paid at their regular hourly rate, so participants are not motivated by compensation to work faster**.

In the between-subjects experiment, participants were evenly split into three 26-people treatment groups to annotate identical sets of videos. The annotators' profiles ensure similar average experience between the groups. The assignments also considered people's day/night shifts and computer setup to ensure a fair comparison.

The participants have at least two months or up to five years of data annotation experience, with an average experience of 20.9 months. We use the median experience of 17 months to split the user expertise factor so each group has about half novice and half veteran workers (see Table 4.1). All annotators were aware of participating in a study testing new AI-assisted annotation algorithms and were free to leave the study at any time. The Human Subjects Committee (HSC) approved our procedure and each participant was provided a consent form during the survey session.

Group A servers as the baseline, they use an efficient annotation tool that supports linear interpolation [136] but solely relies on manual annotation in both parts of the

study. Groups B and C work with their AI teammates in Part 1 of the study. They use the same tool as Group A but the AI will have pre-annotated the videos (see example in Figure 4.3). Group B reviews the restrained AI recommendations that prioritize precision. Group C reviews the zealous AI recommendations that prioritize recall (see a, b in Figure 4.2). The treatment groups are summarized in Table 4.1 or Figure 4.3. We informed the participants in Groups B and C that they are working with an AI that provides recommendations to assist their annotation work, but they do not know the difference between the two human-AI groups.

4.4.3 Experiment procedure of the two-part study.

Before beginning the study, we organized a video conference training session with each treatment group to calibrate the task background and requirements. All participants were also asked to review the instruction text and a training video on the landing page. Previous pilot study users become supervisors in each group to ensure all participants have finished the training and the surveys before processing to the next step. We also created three instant messaging (IM) groups to answer questions and send out reminders when necessary. The overall procedure can be summarized as follows:

$$\text{Training} \rightarrow \text{Survey 0} \rightarrow$$
$$\text{Part 1 (24 videos, different methods)} \rightarrow \text{Survey 1} \rightarrow$$
$$\text{Part 2 (12 videos, same method)} \rightarrow \text{Survey 2}$$

In **Part 1**, all participants from Groups A, B, and C each annotated 24 videos using different methods. For each annotator, the videos were assigned in random order by the annotation platform. We also reminded all participants to avoid taking breaks longer

than five minutes before finishing a video, so the timing is more accurate. Depending on the method and individual pace, it took all groups on the order of two to three weeks to finish Part 1. In **Part 2**, all participants annotated another 12 videos from similar scenes. But we took away the AI assistance from the two human-AI teams B and C in order to find out if their previous human-AI collaboration experiences trained them in any way so that they would perform differently on manual annotations from here on out.

A post-task survey was administered after each part of the study. **Survey 0** was set to "repeat until perfect", this was to verify that the participants were clear about the task requirements before they could start the actual annotation. **Survey 1** focused on getting people's immediate feedback on their experience working with the AI they were paired with. Questions include the correctness and consistency of the AI recommendations, and if the AI made their job easier. This allows us to compare if participants' subjective feelings match the different AI recommendations' underlying personae (high-precision vs. high-recall). **Survey 2** focused on comparing the annotators' preference between AI-assisted and human-only methods after they had experienced both workflows on the same task.

4.5 Results

In this section, we present our study results and analysis by answering each research question presented in Section 4.1. For statistical analysis, we ran one-way ANOVA or one-way Welch ANOVA tests, depending on the underlying assumptions being satisfied, followed by Pairwise Tukey-HSD or Games-Howell post-hoc tests, respectively. To examine interactions between factors, we conducted two-way ANOVAs followed by Pairwise Tukey-HSD or Bonferroni-corrected post-hoc tests. We adopted Type III sums of squares in ANOVA to address unbalanced data.

Research questions **Q1**, **Q2**, and **Q3** focus on results from Part 1 of the study (Figures 4.6, 4.7, 4.9, and 4.11a), in which Groups B and C collaborated with restrained and zealous AIs. Question **Q4** focuses on results from Part 2 (Figures 4.8, 4.10, and 4.11b) to examine how the prior human-AI collaboration experience could affect the users.

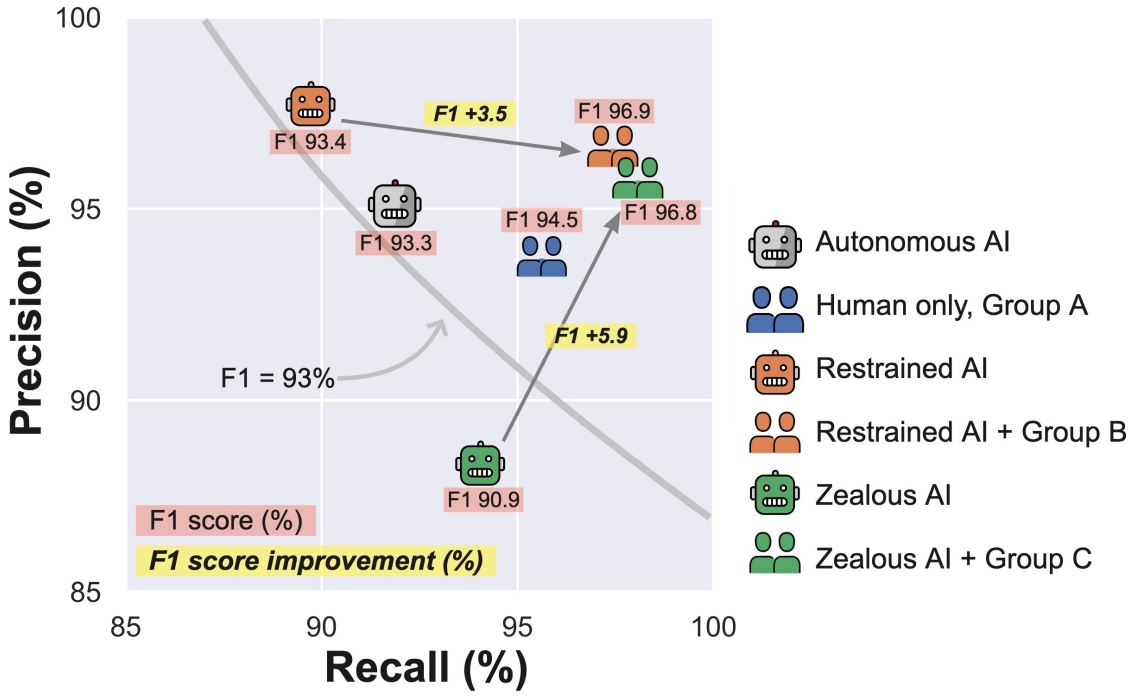


Figure 4.6: Visualizing each group's overall annotation quality on the precision-recall plot with F1 scores (Part 1). Group A manually annotates all videos and without surprise, they are the slowest (Figure 4.7) with a quality better than Autonomous AI alone but worse than the two human-AI groups' team effort. Annotators in Groups B & C had to ***accept***, ***reject***, or ***solve*** the face boxes pre-annotated by the restrained or zealous AIs to improve the human-AI team performance. The arrows show how much humans improved from the AIs' initial annotation.

4.5.1 Q1: Can the human-AI teams achieve "complementary team performance" in this task?

Bansal et al.[93] defines *complementary team performance* as the human-AI team performance exceeding both the human-only and AI-only performance.

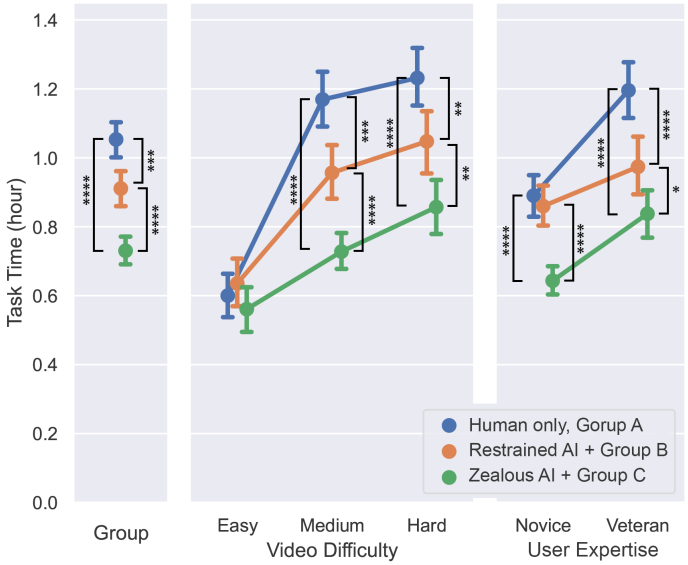


Figure 4.7: Average annotation time for a single video in **Part 1**. Lower is better. Error bars represent the 95% confidence interval. Treatment Group A used a baseline manual method and the annotators in Groups B and C reviewed restrained and zealous AI recommendations in Part 1. Groups B & C included the GPU time used to calculate the AI recommendations.

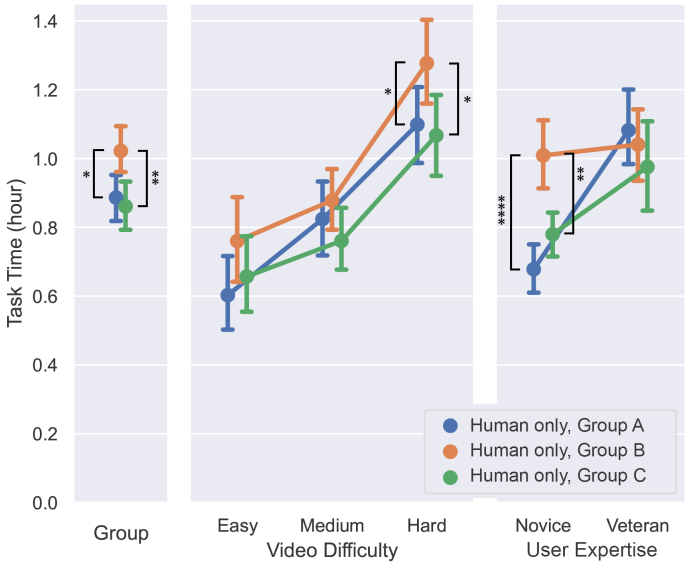


Figure 4.8: Average annotation time for a single video in **Part 2**. After working 2-3 weeks on Part 1, every worker annotated another 12 videos in Part 2 but all used the same manual tool without AI recommendations. We no longer see a significant difference between Groups A & C but Group B is now slower in hard videos, mainly caused by novice workers.

Figure 4.9: The recall distribution of annotated videos in **Part 1**. For the purpose of visualization clarity, we plot the 75-100% range in all recall distributions, which omits maximally 2% of outlier cases. Higher recalls and a "shorter tail" are better. The average recall is marked with a darker diamond. The recall distribution reveals **the likelihood of having a higher quality result**, an insight needed to analyze results from crowdworkers. E.g., in hard videos (right), annotations from "zealous AI + Group C" have a shorter tail than other methods, as expected, the high-recall zealous AI recommendations make it easier for more people to achieve higher recalls especially when people's attention are pushed to the limit when there are three or more faces to track across many frames simultaneously.

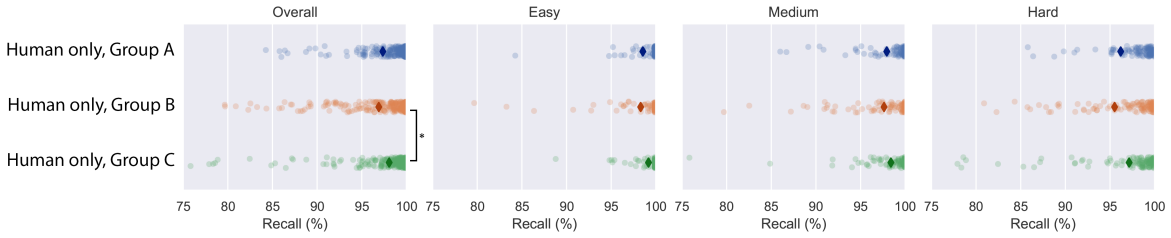
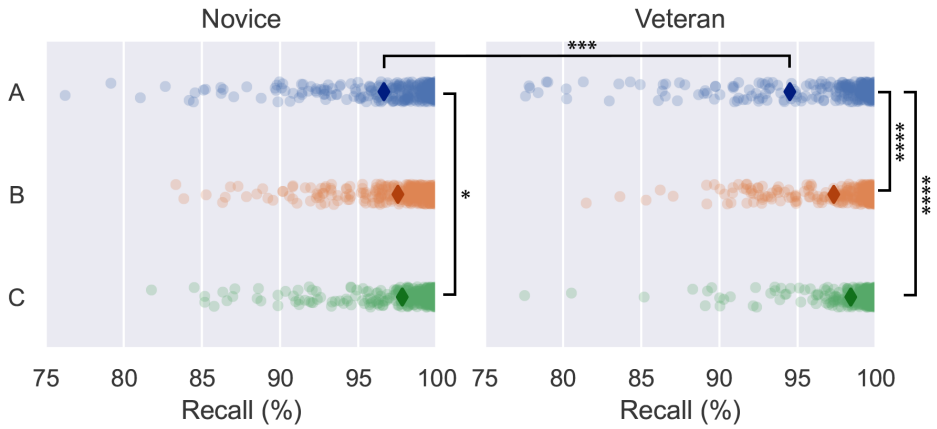
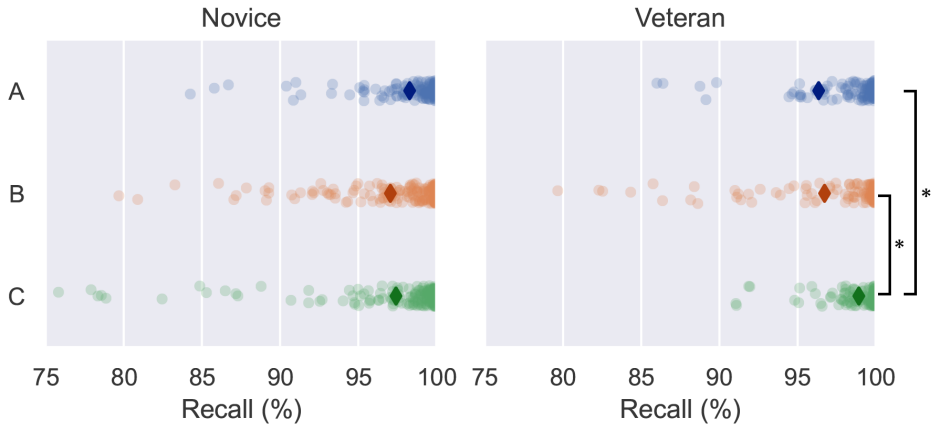


Figure 4.10: The recall distribution of annotated videos in **Part 2**. The previously human-AI collaborative Groups B & C no longer have access to the AI recommendations so they used the same manual method that Group A have been using. The overall subplot (left) shows visible longer tails from these two groups, especially Group C in hard videos (right), indicating a discrepancy in individuals' performance now without the help from AIs.



(a) Part 1 (different methods between groups)



(b) Part 2 (same method: human only)

Figure 4.11: Recall distribution of annotated videos split by user expertise. Figure (a) shows both human-AI Groups B & C gained advantage over the manual method Group A mainly through veteran workers. The longer tails in Figure (b, novice) provide a new perspective to interpret Group C's long tails in Figure 4.10 (Overall) that the performance discrepancy is mostly caused by novice workers after they lost access to AI recommendations.

Figure 4.6 shows the two human-AI teams B & C reached comparable F1 scores of 96.9% & 96.8%, respectively, significantly better than the human-only Group A that reached 94.5% (Welch $F_{2,1151} = 18.2$, $p < 0.0001$). Both human-AI teams improved F1 accuracy and recall significantly compared to their human-only counterpart.

Because the high-stakes nature of this task rules out autonomous AI as a viable option, we really only need to compare the human-AI team performance with human-only performance in Part 1 of our study. However, to verify complementary team performance, we also verify that the two human-AI teams achieved higher performance in terms of F1 scores and recall than their respective AI's initial standalone performance.

Comparing each human-AI team with their perspective AI teammates' initial performance – Group B annotators improved the restrained AI from 93.4% to 96.9% (Welch $F_{1,1228} = 178$, $p < 0.0001$), Group C annotators improved the zealous AI from 90.9% to 96.8% (Welch $F_{1,837} = 169$, $p < 0.0001$). Both human-AI teams improved significantly from their respective AI teammate's solo performance.

It is understandable that Bansal et al. only considered accuracy and did not compare task completion time in complementary performance, since the human-AI teamwork will undoubtedly add more time than AI alone. As we discussed, task completion times directly affect the operation cost as people are paid at an hourly rate, making it a critical metric for annotation tasks, so we additionally compare the human-AI teams' task completion times with the human-only team.

We saw overall significant differences between all three groups on task completion time (Welch $F_{2,1039} = 48.6$, $p < 0.0001$), as shown in Figure 4.7, left. As a baseline, on average it took 1.05 hours for Group A to manually annotate a 30-second video of 900 frames. Group B took a significantly shorter time of 0.91 hours (Games-Howell $p < 0.001$) to review the restrained AI recommendations. Group C only used 0.73 hours to review zealous AI's recommendations, also significantly shorter than the human-only Group A

(Games-Howell $p < 0.0001$).

It is also worth noting that Group C, the zealous human-AI team, had an overall significantly worse starting point than Group B in terms of F1 score: 90.9% vs. 93.4% (Welch $F_{1,854} = 35.32$, $p < 0.0001$) as shown in Figure 4.6. However, annotators working with the zealous AI managed to achieve a significantly higher improvement in F1 score of +5.9% vs. +3.5% (Welch $F_{1,934} = 45.02$, $p < 0.0001$) in significantly less time! This disadvantage for Group C provided the opportunity to demonstrate that our deduction in Section 4.3.3 was correct – a human-AI team can do better in both time and quality (in terms of F1 improvement) by asking the human to ***reject*** more false positives and only ***solve*** the most challenging faces, i.e., the high-recall zealous AI.

In summary, we have not only verified complementary team performance on accuracy, but also showed human-AI teams could achieve significantly shorter task completions time in a real-world case study.

4.5.2 Q2: Which AI helps annotators be more efficient, i.e. save time?

We mentioned that the professional **annotators are paid at their fixed hourly rate in this task**, which means 1) they are not necessarily motivated to work faster, and 2) from the business perspective, their task completion time directly impacts operation costs. We discussed in Section 4.5.1 that overall, both human-AI teams have significantly shortened task completion time compared to the baseline Group A (Figure 4.7 left). Specifically, the zealous AI recommendations help annotators use 20% less time than the restrained AI recommendations with statistical significance (0.73 hours vs. 0.91 hours, Games-Howell $p < 0.0001$).

Video difficulty.

Figure 4.7 (middle) plots task time by video difficulty and saw a significant interaction between group and video difficulty on task completion time (ANOVA $F_{4,1577} = 5.37$, $p < 0.0001$, $\eta_p^2 = 0.016$, small). Specifically, Group C which reviewed zealous AI recommendations used significantly less time than both Group A and B in medium videos (Bonferroni $p < 0.0001$ & $p < 0.0001$), as well as in hard videos (Bonferroni $p < 0.0001$ & $p < 0.01$). But no significant difference was found for easy videos among the three groups.

This observation matches very well with our expectations to different video difficulties: the built-in linear interpolation tool for manual annotation is very efficient in tracking a single face continuously, but **AI recommendations can dramatically reduce task time when tracking multiple faces simultaneously in medium and hard videos**. This finding allows the system designer to optimize efficiency further: if we know a certain portion of the data has one or fewer people in each frame, it would be reasonable to bypass the AI pre-annotation to save on the GPU budget.

User expertise.

When solely considering the user expertise factor, we were surprised that veteran workers are overall significantly slower than novice workers in both parts of the study (Welch, Part 1: $F_{1,1380} = 85.6, p < 0.0001$, Part 2: $F_{1,665} = 22.2, p < 0.0001$)! However, if we consider how people are paid, this result would be a reasonable optimization given the incentives – veteran workers know the acceptable work pace, so they do not need to work faster than necessary. We further discussed worker's incentives in Section 4.6.2.

When we consider the group and user expertise factors at the same time, as shown in Figure 4.7 (right), both novice and veteran workers in Group C who reviewed the zealous

AI recommendations were significantly faster than the baseline (Bonferroni $p < 0.0001$ & $p < 0.0001$), while only the veterans in Group B finished faster (Bonferroni $p < 0.0001$). This allows us to infer that, unlike the restrained AI that helps veterans more, **the zealous AI can consistently improve user completion time for both novice and veteran annotators**.

4.5.3 Q3: Which AI helps annotators achieve higher recall?

From the F1 scores in Figure 4.6 we know that both AI-assisted methods yield significantly higher-quality annotations than the baseline method (compared in Section 4.5.1), yet we saw no clear winner between the two human-AI teams. Because recall is paramount in video anonymization tasks, we analyze Group B and C's recall performance in detail.

Figure 4.9 shows that Group C, the annotators who reviewed zealous AI recommendations, have an overall significant advantage over Group B, which reviewed restrained AI recommendations (Games-Howell $p < 0.01$). Interestingly, we noticed **a visible shorter tail** in Group C's recall distribution in hard videos (Figure 4.9, right). This observation matches the very nature of zealous AI – giving more recommendations, even low-confidence ones, so the human teammate is less likely to miss a face. This strategy is especially effective in hard videos because tracking too many faces simultaneously pushes the user's attention to its limit. **Zealous AI's superfluous recommendations allow the user to focus on the action of *reject*, rather than searching for missing faces and then *solve***.

Taking user expertise into account, Figure 4.11 (a) reveals that while both AIs improved the veterans' recall performance compared to the baseline Group A (Bonferroni A/B: $p < 0.0001$, A/C $p < 0.0001$), for novice workers, we only saw a significant advantage of Group C over Group A (Bonferroni $p < 0.048$). It corroborates our previous

finding on completion time that "the zealous AI can consistently improve both novice and veteran annotators" and extends the statement to higher recalls percentages as well.

4.5.4 Q4: Will collaborating with an AI improve or hurt user skills?

Should the annotators lose access to their AI teammates in the future, how will they perform? While we are interested in improving human-AI team performance, we should also seriously consider how the prior human-AI collaboration experience would affect people's skills in the long run before deploying a new system.

To find out, we removed AI recommendations from Groups B and C in Part 2, so all groups now work with the manual tool that they have always been using for other projects. It took most annotators two to three weeks to complete Part 1 of the study. For the sake of interpreting the results of Part 2, we can consider this period a training period and their performance in Part 2 showcasing the effect of this medium-term training effort.

Both Groups B & C collaborated with their perspective AI teammates for 2-3 weeks, **but the restrained-AI-trained annotators in Group B performed worse than their peers in different ways** – the novice workers were significantly slower than both A & C, especially in hard videos. The veteran workers' annotations had lower recall percentages than the zealous-AI-trained workers in Group C.

Completion time.

Figure 4.8 shows the task completion time of Part 2's 12 new videos without AI recommendations. In all video difficulties, Group C, annotators who previously worked with the zealous AI in Part 1, managed to finish as quickly as Group A, the annotators who were trained using the very manual method now in deployment for all groups. It

shows that training with zealous AI recommendations does not negatively affect users' task completion time on subsequent manual tasks.

However, we were surprised to see that Group B annotators trained with the restrained AI became overall significantly slower than Groups A & C (Tukey-HSD A/B: $p < 0.021$, B/C: $p < 0.01$), and more specifically in hard videos (Bonferroni A/B: $p < 0.044$, B/C: $p < 0.013$). Figure 4.8 (right) shows that the effects stem mainly from the novice users (Bonferroni A/B: $p < 0.0001$, B/C: $p < 0.01$).

Recall.

On annotation quality, Figure 4.10 shows the Groups B annotators, trained by the high-precision restrained AI now produce lower-recall annotations (Games-Howell $p < 0.05$) than Group C which was trained with the high-recall zealous AI. The user expertise breakdown shows the effect mostly comes from the veteran workers (Bonferroni $p < 0.028$).

What caused the negative training effect from the restrained AI?

We would think that annotators in Group B should perform better in Part 2 of the study now that they have to manually annotate – they practiced more on manually adding missing faces (***solve***) working with the restrained AI recommendations. In contrast, Group C which trained with the zealous AI focused on ***rejects***. However, the experiment results show otherwise. Why was only Group B negatively affected? We believe there are two main factors in play:

1) Not optimizing the AI teammate for the human-in-the-loop workflow. Despite the fact that both AIs used the same face-detection model to generate the untracked bounding boxes in each frame for the tracker to process, the restrained AI recommendations were produced by ByteTrack [103] which is designed for autonomous tasks

rather than for human-AI collaboration. We observed various issues using that tracker directly in pilot studies, so we proposed the FPR tracker specifically for a human-in-the-loop workflow with many optimizations with human users in mind (discussed in Section 4.3.3). Given the fact that only novice users became much slower in Part 2 of our study while veterans, who are more familiar with the annotation tool, were unaffected, we strongly believe that the negative transfer effect can be linked back directly to training with the restrained AI.

2) Not optimizing the AI teammate for the task. Recommendations from the high-precision restrained AI are naturally lower in recall than the zealous AI, i.e., the restrained AI missed more faces. Users who worked with such an AI for 2-3 weeks might actually have gotten used to the AI's pre-annotated videos (in Part 1) as "acceptable quality", thus matching their annotation effort with the less optimal recall when working on their own in Part 2. On the other hand, the zealous AI recommendations – the high-recall AI more exhaustively demonstrated all faces that should be annotated, potentially raising the quality standard for the task.

In conclusion, various pieces of evidence from Part 2 of our study showed that despite decent human-AI team performance when working with the AI, naively deploying an AI system into a human-AI setting without considering the nature of the task or without optimizing it for the human teammates could lead to negative effects and potential deskilling of the users.

4.6 Discussion

4.6.1 The key to forming a strong human-AI team

We propose the restrained AI and the zealous AI to depict the tradeoff between precision and recall as two characteristics that have the potential of becoming advantages in human-AI teams if used properly. By actually using the annotation tools and watching annotators' screens for many hours, we observed that annotators need much less effort in improving precision than recall in a model-assisted annotation task, i.e., rejecting an incorrect box is much easier than adding a missing box, thus we should delegate more effort in improving recall to the AI so human only handles the most difficult boxes that the AI missed (Figure 4.2c).

We think **an important insight from this study is that it is worthwhile to identify the complementary strengths of both human and AI teammates through an in-depth analysis of the task at hand**. While our observations can improve real-world object detection and tracking annotation tasks, in which correcting false-positive errors are easier for human, another task with a higher cost in correcting such errors could lead to different or even opposite optimizations. Working closely with end users can inspire us to decompose the AI's different properties (in our case precision and recall) and turn them into advantages to complement human skills. We hope this study can motivate fellow researchers to rethink existing AI assistance designs or at least the design for other video annotation tasks.

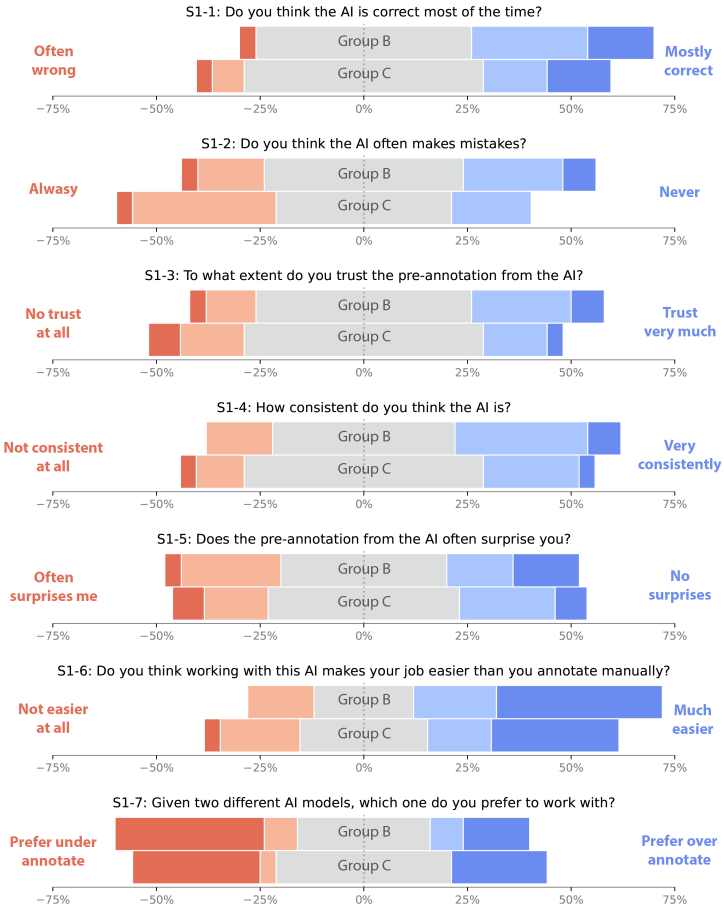


Figure 4.12: Survey 1 (post-Part 1). We normalize each group's five-point Likert scale responses to 100%. 0% indicates no preference. In Part 1's between-subject study, annotators from Groups B & C only worked with a single AI they were assigned to, so we do not compare the responses between B with C.

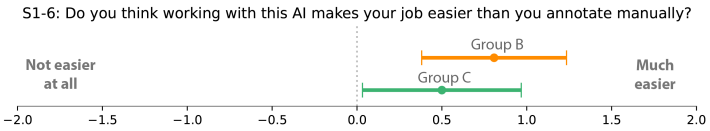


Figure 4.13: Question S1-6 in Survey 1 indicates significant result. The five-point Likert scale responses are converted to [-2, 2] with mean and 95% CI plotted.

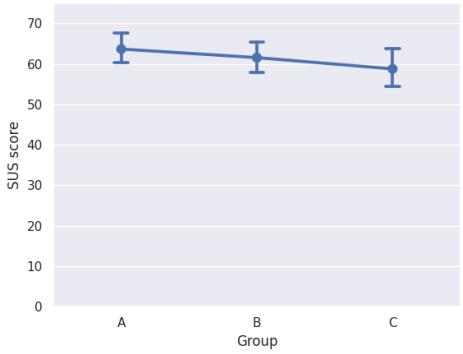


Figure 4.14: A System Usability Scale (SUS) survey was administered at the conclusion of Part 1 of the study. But we saw no significant difference between the groups. Similar to Survey 1 in Figure 4.12, participants tend to provide neutral feedback.

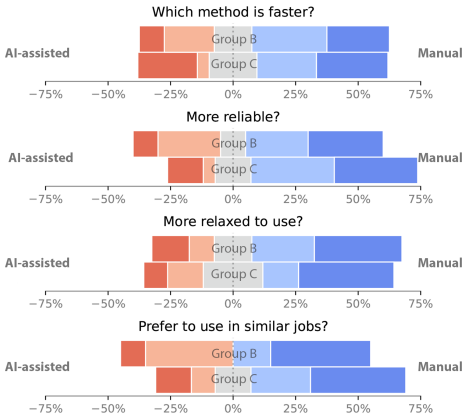


Figure 4.15: Survey 2. Unlike Survey 1 in which annotators answered questions without comparison, Groups B & C have used both AI-assisted and Manual methods at the end of Part 2. Thus this part of the study is close to a within-subject design where the independent variables are the AI-assisted and Manual method.

4.6.2 Can AI teammates set the quality lower bound in a crowdsourcing setting?

We identified and rejected a single veteran user who submitted the majority of the low-quality annotations. This is an unexpected yet not surprising finding in a crowdsourcing setting: when paid at a flat hourly rate, people are not necessarily motivated to work faster. When lacking a quality-based performance evaluation mechanism, people are not necessarily motivated to push for "better-than-sufficient" quality.

However, could there be other users not making an effort in Groups B or C as well but not being identified? Because the two AIs have pre-annotated the videos in decent quality ($F1 > 90\%$), it's hard to tell if someone is actually happy with the AI's recommendations or is not pushing for even better quality.

What we know for sure is that such low-quality submissions, intentional or unintentional, will certainly appear in other real-world crowdsourcing tasks. However, in absence of ground truth, we won't be able to identify them in a real-world setting. It is also very costly to identify bad submissions – ImageNet asks 10 votes for each image [142], and Microsoft COCO asks 3-5 workers to judge each segmentation [99].

Could the AI recommendations have played a critical role in preventing low-quality submissions, i.e., setting a lower bound for the annotation quality? While not verified in our study, this observation could provide yet another strong motivation for human-AI collaboration in a crowdsourcing setting. We encourage fellow researchers to consider this in future experiment designs.

4.6.3 Seemingly contradictory survey results

Figure 4.12 shows user responses to the Survey 1 questions, with each group's five-point Likert scale responses normalized to 100%. 0% indicates no preference. Specifically,

question *S1-6* (Figure 4.13) indicates that users from both human-AI teams, B and C, think that working with the AI makes the task easier than annotating manually. However, in Survey 2 (Figure 4.15), after users have tried both the AI-assisted and the Manual methods on the same task of similar videos, they express higher preference towards the Manual method regarding multiple aspects. As users took each survey immediately after Part 1 and Part 2 respectively, they might prefer the method they just used, but these responses from Groups B & C are in conflict with their continued higher recall in Part 2.

Comparing Figure 4.9 (left) with Figure 4.10 (left), we observe that the Group B & C annotators who had shorter tails in recall distribution than Group A in Part 1 ended up with longer tails in Part 2 after they lost the AI's assistance. It shows that a fraction of low-performing users were apparently held at a higher standard by the AI recommendations, and when the AI teammate was gone, they returned to their preferred standard.

This observation might help explain the higher performance with the AI-assisted method but higher user preference for the Manual method. It also reminds us to take users' incentives into account when designing user preference questions in empirical studies – It is well-known that the most favorable method is not necessarily the best performing method. We administered the System Usability Scale (SUS) survey and saw a trend to support this point in Figure 4.14, but the results are not significant.

4.6.4 Limitations and Future Work

What are the conditions for which our findings hold?

This study investigated a single high-stakes task that met the two aforementioned conditions: 1) either recall or precision is far more important than the other, and 2) the complementary strengths of human and AI can be identified and the precision-recall

tradeoff can be exploited to improve the important metric for the given task. We proposed and observed that delegating more recall effort to the zealous AI can significantly improve team performance, which was mainly motivated by our observation that ***reject*** is much easier than ***solve*** for humans in AI-assisted annotation. Will our findings still hold if ***reject*** is easier than ***solve*** in a different task? What about precision-demanding tasks? We would love to see more HCI and AI researchers conduct latitudinal studies in multiple recall- or precision-demanding tasks to test and refine our findings.

Tasks without high-performance models.

Face detection is a well-studied problem with high-performance AI models. While we showed in Figure 4.6 that the AI and human can reach similar performance in this task to achieve complementary team performance, will our findings stand if either the human's performance or the AI's recommendations are much worse than the other? What is the lower bound F1 score limit for either the human or the AI to maintain complementary team performance? What are the F1 or precision/recall conditions for other researchers to reproduce our findings?

Limitation from data and participants.

We used a subset of realistic, egocentric video dataset [102] in this study to measure with the skill of locating faces – a human instinct that comes with relatively small inter-personal differences. However, could our findings still play a major role if the task was to identify and track other objects that could have larger inter-personal differences? Furthermore, working with amateurs via crowdsourcing platforms would introduce larger variances between individuals than with the professional workers employed in this study. Researchers would need to put more effort into benchmarking or measuring the human factor in such follow-up studies.

Incentives for users to actively perform better.

We discussed in Section 4.6.3 observations that methods with better performances are not necessarily favored by the users. I.e., the users were involuntarily pushed to have higher performance by their AI teammates. From a system designer's perspective, the AI teammate should help users to voluntarily perform better given the right incentives.

4.7 Conclusion

In this work, we look beyond the accuracy of AI recommendations to explore a new direction to improve human-AI team performance – the tradeoff between precision and recall in model tuning. We propose the concept of restrained and zealous AIs for high-precision and high-recall recommendations and conduct an experiment with 78 professional annotators to compare if and how the different AI recommendations can affect team performance in high-stakes human-AI collaboration. This work serves as a new example of complementary team performance in a large-scale realistic setting.

An in-depth analysis of the task helped us identify an optimization opportunity to harness complementary human and AI strengths utilizing the tradeoff between precision and recall in the AI model tuning – given the importance of recall in face anonymization and the higher cost for humans to improve the recall in video annotation. We showed that the proposed high-recall zealous AI helps annotators achieve significantly better performance than the high-precision restrained AI in the video annotation task. Our follow-up study removed AI assistance and observed potentially negative training effects to the users – if an AI is naively paired with humans without optimizing it for the task at hand or for the human-AI workflow. We feel these findings have important implications for the design of AI assistance in recall-demanding scenarios. We hope this work can also inspire researchers to look for additional directions in model tuning to improve human-AI

team performance.

4.8 Acknowledgments

This work was partially done during the first author's research internship at Appen. We thank all anonymous reviewers for their insightful comments and suggestions. We thank Huan Liu for her support and hand-drawn figures, Yue He and Yuedong Wang for their time and discussion, and members of the UCSB Four Eyes and Expressive Computation Laboratories for their helpful feedback. This work was partially supported by ONR awards N00014-19-1-2553 and N00014-23-1-2118.

Chapter 5

In-Situ Learning for User-Guided Personal AI

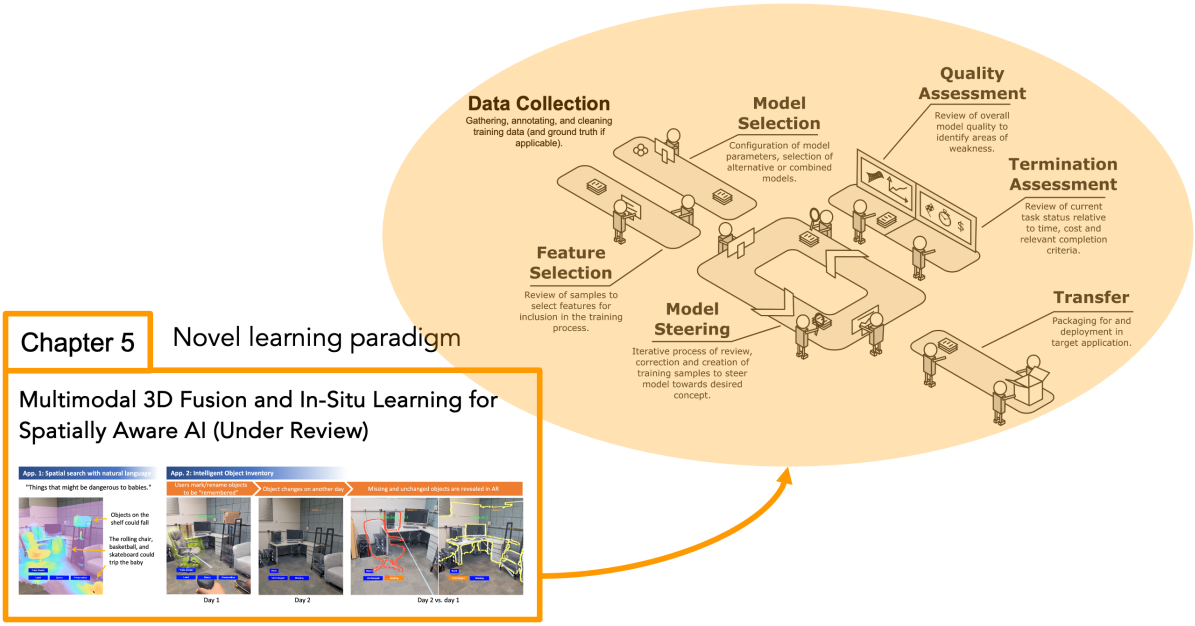


Figure 5.1: Combining what we have learned, we propose "in-situ" learning that aims to take the user's input during every stage of the model shaping process to produce personalized AI models.

In Chapter 2 and Chapter 3 we have shown that AI system designers can help improve the utility of end-to-end models by working closely with users to identify gaps and opportunities for user input and design interfaces with the target users in mind. In Chapter 4, we showed that sustainable human-AI teaming requires a thorough understanding of the team's complementary strengths and the priorities of the task.

In this chapter, combining insights from previous works, we propose a novel approach to real-world human-AI collaboration. We introduce a different machine learning paradigm called "in-situ" machine learning, which encodes real-time user data into models for personal knowledge storage and for human-AI teaming assistance.

5.1 Motivations for in-situ machine learning

We discussed in Section 1.2 that in an ideal interactive machine learning workflow, end users should play a central and active role by providing training data (e.g., domain knowledge or user preference) to the model, reviewing the model's current state, and taking appropriate action (e.g., more training samples or stop the training) in an iterative fashion. This is because highly personalized AI models are preferred or required in certain situations, and the AI model's end users, or their satisfaction with the AI's performance, are often the critical benchmark for such personalized models. Thus it is important to identify the situations in which user-guided personalized AI models are desirable, which help clarify the motivation and best use cases for "in-situ" learning:

- **The model in need does not exist.** Today's computer vision AI models are still predominantly trained for specific tasks, following the generalized workflow outlined in Section 1.2. For domain experts, such as physicians or astrophysicists, who possess specialized data and the knowledge necessary to annotate the date or have unique task objectives that mainstream AI models have not yet accommodated,

suitable models often do not exist. In such cases, providing an accessible means for end users to train or personalize AI models using their own data and tailored objectives is essential.

- **The generic model performs poorly.** For common tasks that utilize generic models, the performance of these models may vary due to the inherent limitations of training on fixed, albeit large, datasets. For instance, a generic scene understanding model that excels in benchmark scenarios might struggle with the diversity and complexity of real-world environments in augmented reality (AR) applications.

- **Personalize AI models for individual users.** Users' inputs and demands for the same task can vary between individuals. This variability greatly affects users' satisfaction with the AI models. In such scenarios, generic models often do not adequately meet personal needs or preferences, even if the model is capable of performing the task. For such scenarios, we propose to take advantage of the output from other computer vision models or the compact and meaningful representations from foundation models to personalize AI with real-time training guided by the end users. We will demonstrate several such use cases later in this chapter.

- **The need for data privacy.** As AI models grow more powerful and larger, it is now common for some models to be hosted on GPU servers and only accessible over the internet. Transmitting user data to online servers present risks to user's privacy. Thus, the small, personalized models that we propose to train locally and operate offline provide valuable data security for certain applications.

"In-situ" is a Latin term meaning "in position" or "on-site." The "in-situ" nature of our proposed concept is characterized by the following observations for human-AI teaming in real-world tasks:

- **Raw data.** We discussed in Section 1.2 that in conventional machine learning workflows,large and diverse datasets are collected for end-to-end training. However, as we live in a complex and constantly evolving world, the offline-collected, static datasets can become outdated and fail to reflect our dynamic or personal environments. To address this observation, we propose to collect **dynamic training data** at the location and at the time when the user interacts with the AI in the environment for the specific task.

- **Ground truth.** End users' individual preferences are often not captured by the generalized consensus in large-scale datasets. Furthermore, user preferences and their perceptions of the environment can be moving targets that change over time, which is why we emphasize the **user-defined ground truth** in in-situ learning that gathers individual's most recent preferences, defines the model for different users, and changes over time.

- **The model.** AI models evaluated on fixed benchmarks are models optimized for static datasets, not for the end users. These models will suffer the same biases or constraints as the static datasets on which they were trained. Similar to interactive machine learning, in-situ learning values user satisfaction over model performance. Thus, **an in-situ model's performance is evaluated by user satisfaction** instead of by fixed benchmarks. It's at the user's discretion to decide if the model is sufficiently optimized, or else they can provide more data to continue the training or provide user feedback to fine-tune the model.

The above use cases and observations motivate us to propose a novel interactive and online machine learning paradigm for real-world human-AI teaming, termed "in-situ machine learning." This concept represents a system design philosophy where real-time user-collected data, such as the task environment and user input, is encoded into a personalized neural network in real time. This network functions both as a repository of the user's accumulated knowledge and as a decision-making assistant tailored to support real-world human-AI teaming. This approach aims to enhance the relevancy and efficacy of AI support in complex, real-world settings by ensuring that the learning process is closely integrated with the user's current context and feedback.

It is important to mention that we do not propose in-situ learning to replace conventional AI models designed for general use cases, i.e., the models for everyone. In fact, in several real-world demos in this chapter, we take advantage of the prediction outputs from generic pose estimation models, vision-language models, foundation computer vision models, and their compact latent representations to optimize a much smaller in-situ model as the personalized AI for individual users. In-situ learning serves as the user-facing AI that aims to extract and store user's knowledge or preferences for later assistance.

5.2 In-situ learning proof-of-concept prototypes

In the remainder of this chapter, we will demonstrate four real-world use cases for the proposed in-situ learning, including proof-of-concept prototypes and a prototype system for spatially aware reasoning. These diverse applications adopt the in-situ learning design philosophy and utilize various computer vision models to train highly personalized AI systems:

Section 5.2.1 A pose estimation model that learns a user's unique poses in less than a minute and can be used as a personalized physical therapy training assistant;

Section 5.2.2 Flexible segmentation models that can learn user-defined abstract concepts with simple strokes as the interface for user input and model guidance;

Section 5.3 A full-fledged augmented reality system that utilizes in-situ learning for AI-powered reasoning involving physical environments.

5.2.1 In-situ learning for personalized pose detection

Why is in-situ learning beneficial in this use case

The author of this dissertation, who relies on physical therapy (PT) training to manage knee pain, found it challenging to consistently maintain his training at home due to the complexity of remembering numerous poses and simultaneously tracking the accurate poses and timing for each session. On the other hand, while pose estimation AI models are becoming more accurate than ever, similar training applications only support the classification/detection of a fixed number of pre-trained poses. In addition, existing applications that use pose estimation perform poorly based on the author's experience, and they often lack the ability to register personalized PT exercises.

Compared to the pre-trained generic pose prediction models, the key challenge in this task is to learn personalized pose classification models that can capture an individual user's unique physical conditions and constraints. For instance, given the same exercise pose, the different heights or movement limitations (e.g., from injuries) between users can lead to very different pose estimation results. Pre-trained models can fail to correctly classify the pose because of these interpersonal differences.

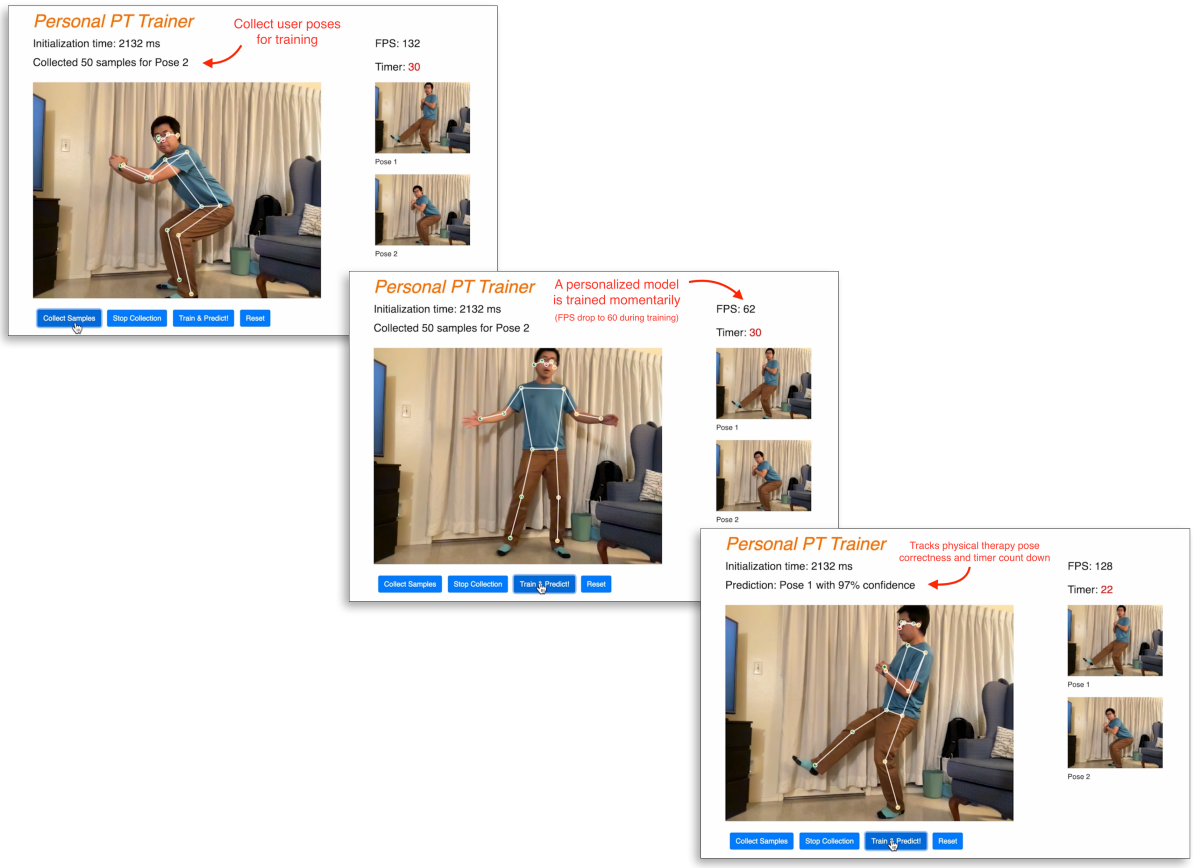


Figure 5.2: The in-situ learning workflow and interface for a personal physical therapy (PT) trainer application. From left to right: 1) The user demonstrates the target poses for real-time, personalized data collection; 2) Without requiring any knowledge of machine learning, an in-situ model is trained in the user's browser in 20 seconds; 3) Once trained, the system checks if the user's pose matches one of the earlier PT training demonstrations, and tracks the progress with a timer.

What can in-situ learning offer in this use case

As an engineer keen on solving real-world problems, the author developed a prototype of a PT trainer application with in-situ learning driving its underlying human-AI teaming components. This application not only assists the user with the exercises at home but also serves as a proof of concept for the in-situ learning idea.

The interface and workflow of the application are illustrated in Figure 5.2. As the user demonstrates each of the specific exercise poses, their body joint information is produced

by a pose estimation model, which serves as the training data for the personalized pose classification model. When trained on real-time collected pose information that considers a user's physical attributes and limits, the in-situ model learns to classify the particular poses just demonstrated by the user. However, the same pose needs frequent updates as the user's physical constraints can improve over time. Another advantage of the in-situ model is its ability to gradually adapt to the user's progress over time. Compared to conventional pre-trained models that require careful finetuning, the low data collection and training cost makes it easy for users to redefine their models based on their satisfaction with the AI, based on the visualization and prediction feedback.

The application is written in JavaScript and deployed in browsers. The in-situ model adopts a simple multi-layer perceptron design. Thus, it can be easily optimized on a consumer laptop in just 20 seconds. The user needs no knowledge of machine learning to train a personalized model. Once trained, the timer starts and stops based on detecting the targeted pose to achieve the PT trainer's assistance. Furthermore, suppose the model behaves poorly in the future (e.g., different lighting, clothing, or improved physical limits). In that case, users can easily update the model by demonstrating the poses again.

5.2.2 In-situ learning for learning abstract concepts

Why is in-situ learning beneficial in this use case

Classification, detection, and segmentation are common computer vision problems that generally follow the end-to-end training workflow we discussed in Section 1.2 – machine learning models are trained and evaluated on a static dataset that is offline collected and annotated. In addition to the issues we discussed in the three observations on the "in-situ" nature of our proposed method (Section 5.1), training on the pre-annotated datasets optimizes the model to the predefined object classes, while users play little role

in providing their insights or preferences in shaping the AI model they need for less common tasks.

Producing a robust, generic classification, detection, or segmentation model that works for most scenarios comes at a high cost in both the data collection and model optimization stages and it requires knowledge and experience in machine learning. In other words, it is not practical or easy for the end user to make their own unique models that detect or segment uncommon objects with conventional approaches. Thus, an accessible means that allows end users to train personalized models is needed.

What can in-situ learning offer in this use case

In this use case, we demonstrate how in-situ learning can help end users who have no knowledge of machine learning to quickly train a segmentation model for any arbitrary concepts with only a dozen images and a few strokes. We show in Figure 5.3 a proof-of-concept application that reads a video file for users to query and visualize objects and concepts with natural language. This is made possible as we preprocess the input video frames with CLIP [143], a multimodal model that maps both vision and language input into the same latent space. By comparing the cosine similarity between the vision embeddings and text query embeddings, we can visualize CLIP feature's native segmentation performance.

Figure 5.3 (1, 2, 3) show that it is possible to achieve open-language detection or segmentation on images through vision-language embeddings, while the probability heatmap visualization (bottom-right of the interface) is not accurate even for simple concepts such as "rabbit," "horse," or "unicorn." In Figure 5.3 (4, 5, 6), our interface allows users to draw simple strokes to highlight objects or concepts. The unicorn toy is a challenging concept as it sits between two similar concepts of rabbits (similar in appearance) and horses (similar in semantics).

In this binary segmentation example, users annotate objects with green strokes for positive samples or red strokes for negative samples to guide the in-situ model to learn the concept, as shown in Figure 5.3 (4, 5, 7). Specifically, in subfigure (5), the overfitted model falsely detected both the unicorn and the rabbit, since they are both white toys, so the user can naturally provide a negative example of the rabbit with a red stroke. Subfigure (6) shows an optimized in-situ model that identifies a much more accurate unicorn than the unoptimized CLIP features.

Another strong advantage of in-situ models is their flexibility. The user can fine-tune the model instantaneously to learn another arbitrary concept. For instance, in Figure 5.3 (7), we add the additional concept of "wall" to the model that is already optimized for unicorn. With just a few more positive and negative strokes, we can train a new model that detects the arbitrary concept of "unicorn + wall" in subfigure (8).

In conclusion, the binary segmentation prototype powered by CLIP embaddings [143] and in-situ learning demonstrates the important characteristics of real-time data collection, user-defined ground truth, and user-satisfaction-based model evaluation. In the next section, we present an augmented reality system that adopted the similar in-situ learning philosophy in facilitating human-AI teaming in an intelligent object inventory AR application.

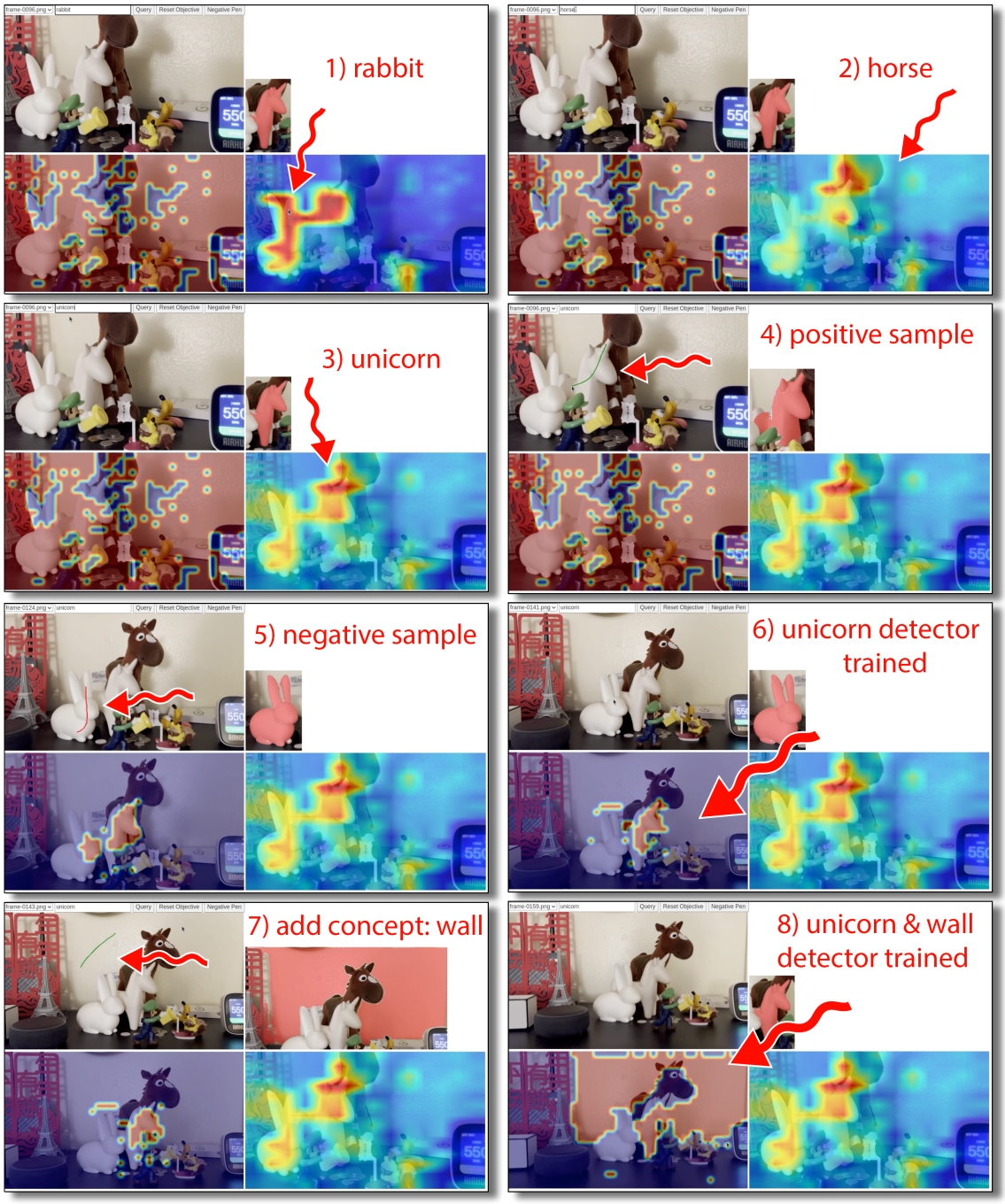


Figure 5.3: Training a segmentation model for arbitrary concepts with in-situ learning. Here we show a new segmentation model for unicron, a rare concept for existing segmentation models, or even for an arbitrary concept like "unicorn + wall" which can be trained with only a few interactive strokes from the user. Details of the workflow are described in the text.

5.3 Multimodal 3D Fusion and In-Situ Learning for Spatially Aware AI

Contents in this section are part of a paper submission currently under review. Authors include C. Xu, R. Kumaran, N. Stier, K. Yu, and T. Höllerer.

Seamless integration of virtual and physical worlds in augmented reality benefits from the system semantically "understanding" the physical environment. AR research has long focused on the potential of context awareness, demonstrating novel capabilities that leverage the semantics in the 3D environment for various object-level interactions. Meanwhile, the computer vision community has made leaps in neural vision-language understanding to enhance environment perception for autonomous tasks. In this work, we embed both semantic and linguistic knowledge into the geometric scene representation, enabling user-guided machine learning involving physical objects. We first present a fast multimodal 3D reconstruction pipeline that brings linguistic understanding to AR by fusing CLIP vision-language features into the environment and object models. We then propose "in-situ" machine learning, which, in conjunction with the multimodal semantic representation enables new tools and interfaces for users to interact with physical spaces and objects in a spatially and linguistically meaningful manner. We demonstrate the usefulness of the proposed system through two real-world AR applications on Magic Leap 2: a) spatial search in physical environments with natural language and b) an intelligent inventory system that tracks object changes over time. We also make our full implementation and demo dataset available to encourage further exploration and research in spatially aware AI.

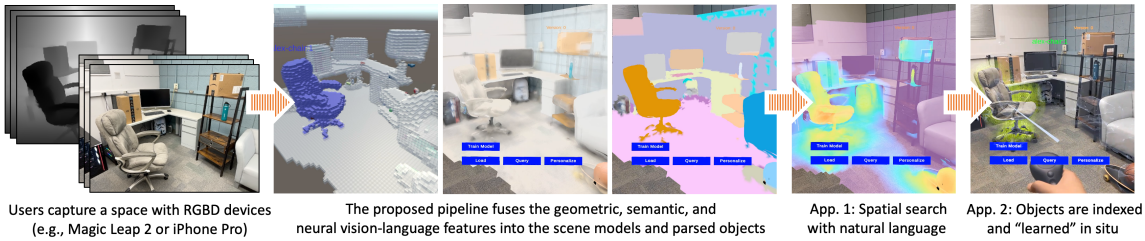


Figure 5.4: We propose a multimodal 3D reconstruction pipeline that prepares physical spaces for vision-language perception and object-level interactive machine learning. Within a few minutes after a user scans the environment, they can search the space with abstract natural language queries or interact with physical objects for spatially aware AI applications through novel AR interfaces.

5.3.1 Introduction

3D scene understanding of physical environments is crucial for context-aware augmented reality (AR). Research and industry endeavors have been steadily advancing the sensing and sensemaking capabilities of mobile computational platforms [144, 145]. Modeling and understanding basic geometric configurations such as room size, solid surfaces, and occlusions enable realistic virtual content placement and interactions [146, 147, 148]. A good semantic understanding and 3D segmentation can reveal the what and where of common objects in an environment to enable complex interactions and deeper blending of the virtual with the physical [149, 150, 151, 152, 153, 154]. Leveraging the power of recent large multimodal models (LMMs) and large language models (LLMs), we can even perform simple spatial and linguistic reasoning in complex real-world scenes [155, 156].

If we continue to push the envelope, what new forms of scene understanding and reasoning are in store for context-aware AR and its applications? We believe a promising holistic approach to probe this direction is to build and test a pipeline that integrates the geometric, semantic, and linguistic information of a real-world environment into its 3D environment model and every object in that environment. To this end, we implemented a TSDF-based [157] 3D reconstruction and segmentation pipeline that fuses neural vision-language features (e.g., OpenCLIP [158]) from AR input frames into the

3D representations of the physical space and individual objects.

The 3D fusion of neural vision-language features automatically enables linguistically meaningful spatial computing. While previous AR spatial search tasks are constrained by limitations of close-set detection or segmentation models, it is now possible to search for arbitrary objects, or even respond to abstract natural language queries in a physical space. We show in Figure 5.4 and Figure 5.7 Application 1 that a heat map responding to the query "Things that might be dangerous to babies" highlights the most probable areas through the AR headset. The physical environment, imbued with vision-language features, can provide valuable information about itself via AR interfaces.

We fuse context-relevant vision-language features into the 3D scene at the abstraction levels of scene voxel, vertex, and individually segmented physical object. Such multimodal fused and semantically indexed objects create more intelligent "virtual twins" than previous 3D representations, and become even more powerful when we add user-in-loop interactive machine learning controlled by AR interfaces. The user's interactions with physical objects provide valuable model steering instructions to train a personalized machine learning model, e.g. for intelligent inventory management.

Imagine a space-constrained cluttered environment full of mission-critical tools, equipment, and notes, such as the International Space Station. A spatially-aware-AI AR companion could help astronauts keep tabs of all the items being used and stowed by the various crew members. It could provide assistive AR guidance for navigation, search, and task support, providing essentially a physical space version control system.

A good use case for demonstrating the advantages of pairing the proposed multimodal 3D fusion with a user-guided machine learning mechanism, which we dubbed "in-situ" learning, is to track the changes of physical objects in the real world. Unlike a conventional version control system (e.g., Git [159]) that can compare the difference between two saved states, if we move a specific red coffee mug from one desk to another, the

simple spatial translation alters an object's orientation or volumetric representation, yet no change would occur semantically. In order to provide users with useful information, the item needs to be identified as the same entity. In office spaces where shared objects rarely stay at the same place or orientation, naive mesh comparison only produces noise. What is needed is a user-trainable classifier that can learn to remember arbitrary physical objects and quickly optimizes for users' changing needs and the task at hand.

To this end, we present a proof-of-concept intelligent object inventory system, demonstrating the ability to remember and re-identify objects in physical environments enabled by multimodal 3D fusion and in-situ learning. The demos run on Magic Leap 2 and work in complex real-world spaces. Once trained with simple user guidance, the system can reveal objects that are missing or remain unchanged over time in a tracked space. In Figure 5.7 Application 2, we show that when a colleague's rolling chair was removed from the tracked scene, we can travel back in time to reveal the disappeared chair at its previously recorded location.

Our contributions can be summarized as follows:

- We present a custom multimodal 3D reconstruction workflow that fuses neural vision-language features into the 3D volumes of the environment and automatically segments objects, unlocking novel context-aware AR interfaces for physical environments and objects.

- We demonstrate the enhanced effectiveness of the fusion pipeline when coupled with in-situ learning in real-world spaces with two novel AR applications on Magic Leap 2: a) spatial search with natural language and b) an intelligent inventory prototype that can track physical object changes.

- We share system design details and open source our implementation and example dataset to help fellow researchers develop future spatially aware AI applications

based on our system.

This paper is organized as follows: Section 5.3.2 discusses the related work. Section 5.3.3 describes the design decisions and technical details of the fusion pipeline, in-situ machine learning, and the scene manager, which connects the two components with the AR headset user. Section 5.3.4 demonstrate two real-world application scenarios of the proposed system.

5.3.2 Related Work

Our work is broadly inspired and germane to topics in mixed reality, computer vision, machine learning, and HCI. In this section, we discuss related work in the areas of AR Scene Understanding, AR Scene Authoring, Physical Interaction in AR, Interactive and Online Machine Learning, Open-Vocabulary 3D Perception, and Version Control for Non-traditional Media.

Augmented Reality Scene Understanding

Scene understanding gives semantic meanings to the reconstructed 3D models of the physical environment, allowing AR headsets to know not only the geometry but also the what and where of objects in the space, which is needed to unlock context-aware AR and interactions. SLAM++ [160] generates an object-level scene description relying on prior knowledge. However, the requirement of a library of known objects prevents this system from generalizing to arbitrary scenes. "FLARE" [146] generates AR object layouts consistent with the geometry of the physical environment. SnapToReality [147] aligns virtual content to real-world 3D edges and surfaces. The Spatial Mapping and Scene Understanding APIs in Microsoft HoloLens [148] and similar AR APIs can infer semantic surfaces such as walls, floors, platforms, and ceilings.

Recent leaps in computer vision, foundation models, large language models (LLMs), and large multimodal models (LMMs) provide new directions in tackling object-level scene understanding tasks. Chen et al. [161], PanopticFusion [162], and Panoptic Multi-TSDFs [163] went beyond geometric-based AR by combining semantic segmentation with dense 3D reconstruction to achieve object-level 3D understanding, removing some of the constraints that SLAM++ was beholden to. Retargetable AR [152] builds a 3D-directed graph characterizing the scene context, such as the location and orientation of objects, so that virtual content can be placed and interactions with the physical environment become more realistic. Pucihar et al. [164] explore how machine learning techniques can automatically detect, recognize and segment scene objects in an intelligent way that allows the system to annotate unprepared environments automatically.

In a more challenging open-vocabulary setting, Yoffe and Sharma proposed OCTOPUS and OCTO+ [165, 156] to automatically place arbitrary objects on the most suitable surface in AR. They chained up a series of state-of-the-art ML methods to build a Mixture of Experts System. They used Segment Anything Model (SAM) [166] to identify individual objects, CLIP and clip-text-decoder [143, 167] to generate object labels, and ViLT [168], CLIPSeg [169], Grounding DINO [170] to verify object guesses. Like many mixture of experts systems, they used LLMs or LMMs such as GPT-4, GPT-4V [171], and LLaVA [172] as the "brain" to reason about the appropriate location for object placement based on the curated text and image inputs from various upstream models.

Compared to their work, which focuses entirely on the question on how to place content in 2D image frame observations of 3D scenes, we tackle the much more general problem of embedding the semantic features within the 3D geometric representation, enabling additional levels of spatial reasoning.

Augmented Reality Scene Authoring

Authoring tools complement automatic scene understanding by allowing content creators or end users to assign semantic meaning or information to the reconstructed scene even in absence of fully reliable automatic detection and segmentation. Early authoring systems relied on designers to manually attach information to the environment. In Columbia's "touring machine" [144] the information augmenting the university campus with building and department information was manually placed, whereas the follow-up MARS system utilized offline and online authoring tools[173]. Mann [174] overlaid text on certain recognized objects in head-worn displays. The "WorldBoard" vision [175] aimed for planetary-scale information placement.

Immersive authoring [176] in AR allowed users to parse things or objects from the reconstructed 3D model while moving around in the actual physical space. SemanticPaint [177] is an interactive online system that continuously learns from the user's segmentation input to predict object labels for new unseen voxels as the user captures the environment. Semantic Paintbrush [178] tackles a similar problem with novel hardware and software solutions that take the user's precise input from a laser pointer. SceneCtrl [150] combined Hololens' plane detection with user input to identify objects for flexible and plausible scene editing. Huynh et al. proposed In-Situ Labeling [179] to facilitate more effective language learning in AR settings. HoloLabel [180] provides a user-in-the-loop 3D labeling system on HoloLens.

Unlike the above scene authoring tools that require careful interactive operations on voxels or meshes, this work has individual physical objects segmented and labeled automatically during the 3D reconstruction process. Users interact with the densely labeled scene at the object level by directly selecting individual objects for personalization, or merging multiple semantically meaningful mesh blocks for correction, without having to

deal with voxels or meshes.

Interacting with Physical Objects in AR

Going beyond geometric-based AR, recent context-aware AR focuses on novel interfaces that interact with the virtual twins of physical objects in the environment, creating illusions that tightly blend the physical and the virtual. Annexing Reality [149] uses physical objects as proxies for virtual content to reduce the visual-haptic mismatch. SceneCtrl [150] on the Microsoft HoloLens provides coarse part-based object selection so users can select, delete, move, or copy virtual twins of real objects to manipulate the physical environment as a virtual but visually plausible scene. Remixed Reality [151] applied a taxonomy of spatial, appearance, viewpoint, and temporal manipulations and interactions on a live 3D reconstruction of the environment, captured by multiple depth cameras. RealitySketch [153] captures user sketchings to create virtual elements bound to physical objects and dynamically respond to real-world changes. TransforMR [181] can replace real-world humans and vehicles with pose-aware virtual object substitutions to produce semantically coherent MR scenes. Kari et al. [154] demonstrated the concept of Scene Responsiveness which maintains visuotactile consistency in situated MR through visual illusions that hide, replace, or rephysicalize real objects with virtualized objects and characters.

Our work is related to the above research as we provide novel interfaces for users to interact with the physical room, in our case through natural language and by tracking physical objects' changes over time. It differs from the above works in that we integrate deep vision-language features into the 3D models to automatically identify (segment and label) and remember individual objects for the application scenarios.

Interactive & Online Machine Learning for AR

Interactive ML and online ML are distinct learning paradigms that often go hand in hand in real-world interactive AR/MR applications. In these applications, training data becomes available in the form of a stream during the user's interaction with the environment.

To learn from previous user gestural and verbal input to predict the segmentation and object labels for new unlabeled parts of the 3D scan, SemanticPaint [177] proposed a streaming random forests algorithm that trains on voxel-oriented patches (VOPs), which are geometric and color features computed from raw TSDF volumes. Semantic Paintbrush [178] adopted the same VOPs as object features (RGB, surface normal vector, and 3D world coordinate) to train a similar streaming decision forests. ScalAR [182] also used a decision-tree-based algorithm to learn from the user's demonstration input.

Unlike previous interactive AR systems that trained decision trees on low-level features, this work utilizes deep vision-language models to generate semantic and linguistically meaningful deep latent features for the environment and individual objects. We combine an object's geometric representation (voxels), appearance (RGB), and vision-language features (CLIP) to produce meaningful object graph representations that are robust against issues that low-level features suffer from, such as changing lighting conditions, orientation, and over-simplified semantics. We do propose our own version of interactive online ML, dubbed "In-Situ Machine Learning" (see Section 5.3.3), to refine and improve the performance of our automatic semantic segmentation and object recognition.

Open-Vocabulary 3D Perception

The attribute "open-vocabulary" describes a system that can recognize objects matching a free-text description, which may contain arbitrary natural-language descriptors (e.g. "a chair whose color is somewhere between blue and green") or abstract concepts (e.g. "what can I use to prop open a door?"). This is a much more flexible, intuitive, and ultimately more useful paradigm in many scenarios compared to the traditional computer vision approach of classifying objects into a pre-determined semantic taxonomy, which is inflexible and cannot be exhaustive.

Several works have presented open-vocabulary systems for 2D image segmentation and understanding, either at the level of patches or entire images [143, 183, 184], while others have focused on dense, per-pixel representations [185, 186, 187]. These systems became possible because of the massive amount of paired images and text available on the internet that were used as training data for vision-language foundation models. In contrast, for 3D data, it is more difficult to directly develop the 3D-language connection, due to the lack of large datasets of paired geometry and text. To address this, a number of works have attempted to bootstrap 3D open-vocabulary perception by distilling or otherwise lifting 2D open-vocabulary models to operate on 3D data [155, 188, 189, 190, 191].

Our system, specifically the *Spatial Search with Natural Language* feature, belongs to this latter category, lifting CLIP features into 3D by back-projecting them into a voxel grid, using a modification of the popular TSDF fusion algorithm. Our system is primarily differentiated by its design to support interactivity in AR. Most importantly, it operates with low latency, without requiring expensive components such as 3D convolutional neural networks or training neural radiance fields. This enables a smooth and familiar AR scanning workflow. In addition, our system builds an implicit surface mesh represen-

tation rather than relying on point clouds or density volumes. This is more amenable to downstream processing and rendering with the traditional graphics pipeline, meaning that it can easily be integrated into existing AR platforms and applications.

Version Control for Novel Media

One of our demonstration applications, intelligent object inventory, tracks certain object changes that are akin to the behaviors in version control systems. Software developers are most familiar with text-based version control systems (VCSs) such as Git [159] that keep track of changes in source code. The research community has explored version control interfaces and techniques on novel media other than text editing. Time-Machine Computing [192] tracks computer desktop states and allows users to visit a previous state. MeshGit [193] proposed a mesh edit distance to measure the dissimilarity between two polygonal meshes in 3D modeling workflows. SceneGit [194] tracks object-level element changes in a 3D scene as well as finer granularity changes at the vertex and face level. The *Who Put That There* system [195] records virtual objects' spatial trajectories from the user's direct manipulation in 3D virtual reality scenes.

VRGit [196] facilitates synchronous collaboration for manipulating and comparing VR object layouts immersively. While the system provides well-defined spatial versioning features, it operates on a library of predefined models because of its VR nature. Research with physical artifacts in mind, such as Catch-Up 360 [197] and works by Letter et al. [198] focused on the changes of a single object rather than room-size environments like in VRGit. AsyncReality [199] used external devices to volumetrically capture physical events for later immersive playback.

Our work differs substantially from the above version control research in that physical objects change in ways different from source code or 3D models – spatial translation, deformation of non-rigid objects, and appearance changes based on lighting conditions.

Naive comparison between two mesh models only introduces counterproductive noises, even if they are perfectly aligned. This work, however, maintains object identity by relying on deep vision-language features embedded in the reconstructed 3D environment. We demonstrate one of the first intelligent inventory systems that automatically track basic object changes (missing/unchanged) in real-world environments in Section 5.3.4.

5.3.3 System Overview

At the heart of our spatially aware AI system that informs the AR user interfaces in this paper is a custom 3D reconstruction pipeline with integrated vision-language fusion and 3D segmentation workflow. The pipeline starts with capturing a physical space – a user walks around with an RGBD device that captures registered RGB images and depth maps to reconstruct the 3D model with geometric, semantic, and linguistic understanding (see Figure 5.4). Figure 5.5 shows the system overview. The accompanying video also demonstrates the interactive possibilities and flow of the system. We will discuss the design of the main components in this section.

Multimodal 3D Scene Model Fusion

Our multimodal scene volume is represented by a multi-channel voxel grid defined over the scene. The channels of this volume are organized into three components: geometry, language, and semantics. The geometric component is a single-channel TSDF volume produced using the typical TSDF fusion algorithm [157] according to the following running-average update rule:

$$D_{i+1}(x) = \frac{D_i(x)W_i(x) + d_{i+1}(x)w_{i+1}(x)}{W_i(x) + w_{i+1}(x)}, \tag{5.1}$$

where $D_i(x)$ is the accumulated TSDF estimate over all past views for voxel x at time i, and $d_i(x)$ is the TSDF estimate for voxel x from the current view at time i. w represents a per-view weight, and W is the total accumulated weight (we refer the reader to Curless & Levoy [157] for further details).

We then propose a simple mechanism to extend the scene volume with additional channels, which are populated by fusing feature vectors from image-aligned 2D feature maps as follows:

$$F_{i+1}(x) = \frac{F_i(x)W_i(x) + f_{i+1}(x)w_{i+1}(x)}{W_i(x) + w_{i+1}(x)}, \tag{5.2}$$

where $f_i(x)$ is a feature vector sampled from view i by perspective projection from voxel x, and F is the generated multi-channel feature volume. The main advantage of fusing features in this manner is that by averaging across views we develop a more accurate multi-view feature and label estimate over time.

We leverage this extension to build the semantic and language components of the volume by fusing in two additional sets of 2D feature maps. The first one (object semantics) is a per-pixel class probability distribution, computed using the panoptic segmentation from k-means Mask Transformer [200]. The second one (language) is a per-pixel CLIP feature computed using OpenCLIP [158]. Since CLIP's feature output for a given image is only a single feature vector with no spatial dimensions, we tile each image into overlapping patches to produce a coarse 2D CLIP feature map. We then define a continuous CLIP feature across the image using bilinear interpolation. While the parameters can vary among capture devices and scenes, our setup resizes input frames to 1024×768 px and uses 256^2 px patches with a stride of 128 px.

Finally, the fusion process results in a per-voxel TSDF estimate, class probability distribution, and CLIP feature, that we use to support downstream applications (Figure 5.5 left). This process exhibits two properties that make it highly amenable to AR

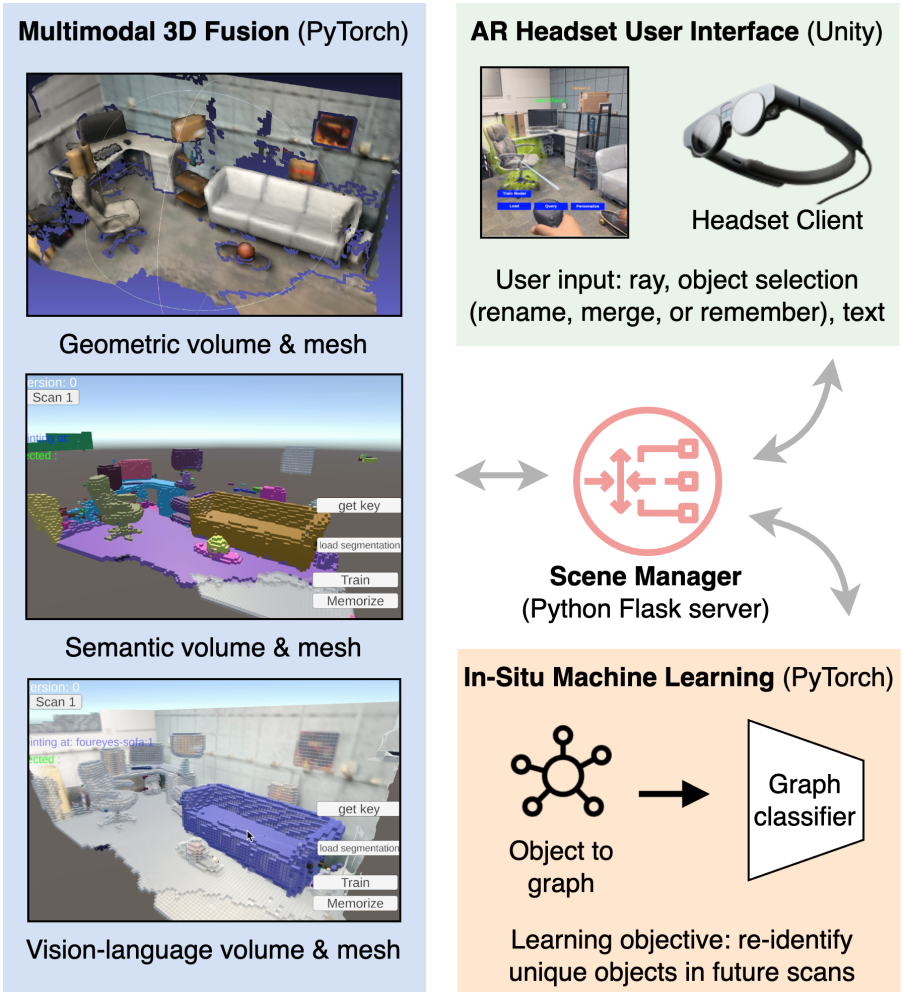


Figure 5.5: System overview.

applications: 1) all three volume components are constructed using a running average update rule, so the process is incremental and can accept new input views at any time without needing to revisit earlier views; 2) no iterative optimization is required, leading to fast online reconstruction. We perform the reconstruction on a local server with a single NVIDIA RTX 3090 GPU, which also trains the in-situ learning model and streams the application results to the AR device for visualization. With our current system design and configuration, it typically takes two minutes to process a $3m \times 3m$ space.

Post-processing and Scene Manager

Following the 3D scene fusion, we perform three post-processing tasks to support downstream applications.

1) Mesh extraction. We run Marching Cubes [201] to extract a triangle mesh from the TSDF volume. This allows for convenient rendering and integration with existing AR graphics pipelines. Figure 5.4 shows various mesh visualizations rendered in AR headsets.

2) 3D semantic segmentation. To create useful spatial awareness for AR, we are interested in going beyond per-voxel semantic information to delineate full objects that users can more easily select and manipulate. We therefore build on the class probability volume developed in Section 5.3.3 by first labeling each voxel with the class for which it has the highest predicted probability, and then segmenting consecutive volumes according to those labels using a custom 3D flood fill implementation. Similar to classic 2D flood-fill algorithms that find connected regions on images [202], our 3D method clusters voxels of the same segmentation class in the 3D volume grid to parse individual objects in the user's surroundings, extracting the complete object boundary, shape, and identity, with no user intervention required (see Figure 5.5 semantic volume).

3) Intelligent object inventory. During the object parsing process, the Scene Manager creates an intelligent object inventory by associating the per-voxel CLIP features with the individual objects. The scene manager, as shown in Figure 5.5, is the central communication hub that a) manages multiple versions of environment models, b) sends and receives data between the AR user interface via HTTP requests, and c) utilizes the in-situ machine learning engine to "remember and re-identify" unique objects for the intelligent object inventory.

After these post-processing steps, users can then easily interact with physical ob-

jects through the virtual pointer on AR devices (see Figure 5.5 user input). Compared to conventional 3D reconstruction focusing on user manipulation of the object model's mesh or volume representation, the proposed multimodal pipeline associated the object identity, metadata, and vision-language features with objects in the intelligent inventory. With CLIP features attached to objects, novel spatially-aware-AI interfaces are unlocked through interactive machine learning, which we will discuss in the next section.

"In-Situ" Machine Learning for Intelligent Object Inventory

We previously defined "in-situ" machine learning in Section 5.1. In this section, we show how it can be used to improve AR experience in complex real-world environments. Since we have attached rich multimodal features to individual objects, one practical optimization objective for the in-situ learning model is to learn to remember and re-identify individual objects across different scans. We define in-situ learning as the process of encoding real-time data into a neural network, such that the network itself serves as both the knowledge container and decision-making unit for downstream tasks, e.g., as a probe to identify changes such as new or missing objects. This is related to AR works discussed in Section 5.3.2 and works that perform online neural scene encoding (Feng et al. NARUTO [203], Sandstrom et al. Point-SLAM [203]), but we introduce the additional temporal dimension to enable differencing across multiple time points (room scans on different days) in the second application scenario.

The "in-situ" (Latin for "in position" or "on site.") nature of our machine learning concept is characterized by the following observations:

1. As we live in a complex and constantly evolving world, the neural vision-language features that represent objects and contexts also change dynamically. These **unlabeled training data** are only generated at the moment when the user captures

the physical space.

2. Novel **ground truth samples** that guide the supervised learning (e.g., personalized object names and merged segments) are generated only when the user interacts with the environment. Unlike offline-collected large-scale datasets that present a universally accepted ground truth of our world [142, 99], the ground truth in in-situ learning varies among users and over time.

3. Similar to typical interactive machine learning, the **model's performance is evaluated by the user** instead of by a fixed benchmark. It's at the user's discretion to decide if the model is sufficiently optimized, or else they can provide more training data by re-scanning the space or re-labeling incorrectly classified objects to fine-tune the model.

4. Also similar to interactive machine learning, the model always immediately reflects the **user's latest annotations and preferences**, which is in contrast to the typical batch processing and incorporation of user feedback necessitated by large offline-trained models.

In addition to the interactive user guidance, we convert an object's irregularly shaped volume representation to a graph representation to optimize for online machine learning. Unlike previous scene authoring AR work that trained on low-level TSDF features alone [177, 178, 182], the multimodal 3D fusion allows us to create a novel graph representation that combines the geometric, semantic, and vision-language features for every object in the scene. As shown in Figure 5.6, we treat every voxel as a node pointing to the object's centroid, which converts an object's irregular voxel representation into a dense graph representation. For efficiency and data augmentation, we stochastically sample 30 voxels from the dense representation in each training iteration to generate a sparse graph,

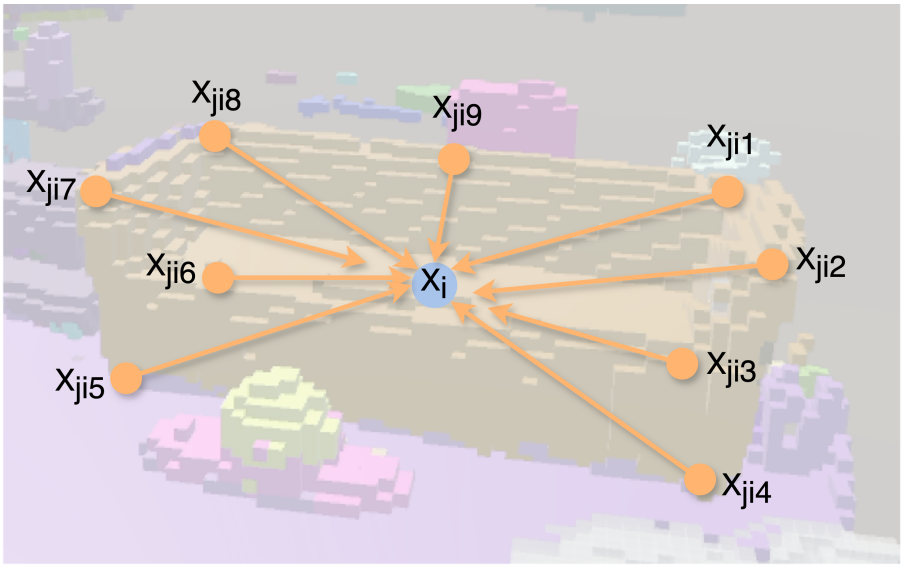


Figure 5.6: During the real-time in-situ training, we sample a sparse graph from an object's voxel representation stochastically, with the voxel location's CLIP feature as the node attribute. This design choice converts the challenging irregular 3D object classification problem into a simpler graph classification problem, which enables us to identify physical objects across multiple scans of the space. The sofa graph above is oversimplified for visualization purposes.

whose node attribute is the voxel location's OpenCLIP [158] vision-language feature, which we found sufficient to re-identify objects and reveal object changes (Section 5.3.4) without having to align the scene models for naive mesh comparison. Additional properties, such as RGB values, the geometric 3D vector pointing at the object's centroid, and relative spatial relationships to other objects can also be integrated as node or edge attributes based on specific task needs.

In other words, we turn a hard 3D object classification task into an easier graph classification task, which maintains its effectiveness even if the object or the environment changes dynamically (spatial translation, non-rigid deformation, varying lighting conditions). Additionally, the in-situ model is incrementally fine-tuned as the user provides new inputs from subsequent scans. Specifically, to learn the graph-based objects, we adopt a dynamic graph CNN [204] as the backbone of the in-situ model to train a graph

classifier that predicts if the graph belongs to a class label previously trained on, or an unknown class (e.g., background objects not marked by user). A single in-situ model is trained for a specific space – it can be tailored by one user for personalization or shared between a group of users for collaboration and information exchange.

To summarize, in-situ learning's novelty lies in the "just-in-time" user-generated data and the evaluation metric that is based on user satisfaction. In real-world applications, system designers should choose the specific type of machine learning paradigm (e.g., supervised or self-supervised), the model architecture, and the training strategy that best supports the task at hand.

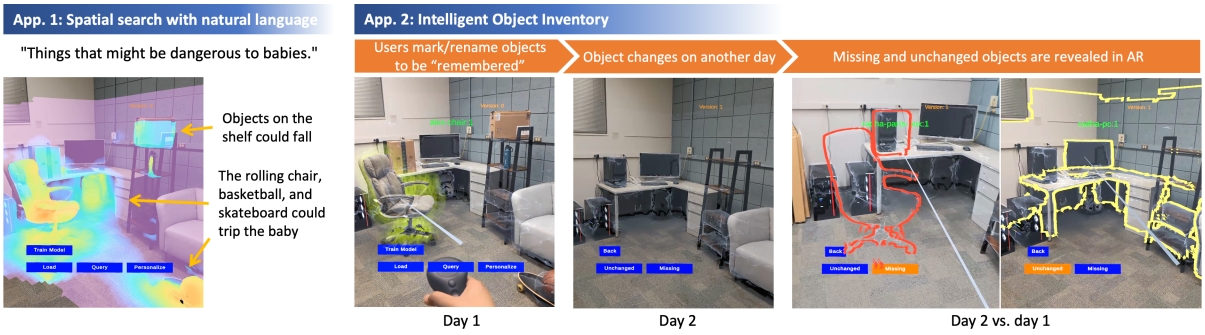


Figure 5.7: Two prototype applications developed on Magic Leap 2 AR headset, demonstrating the potential of the proposed multimodal 3D fusion pipeline and "in--situ" machine learning for real-world scenarios.

5.3.4 Prototype Applications

In this section, we showcase two real-world Magic Leap 2 AR prototype applications to demonstrate the potential of the proposed multimodal 3D fusion pipeline and new tools and interfaces for users to interact with physical spaces and objects when used in conjunction with "in-situ" machine learning.

Spatial Search with Natural Language

We foresee a future of sophisticated real-virtual interactions, requiring deep understanding that goes beyond discrete objects with labels. By fusing 3D vision-language features into the 3D models, our system permits spatial search in a physical space using natural language. The user may issue complex, even abstract queries, for example, "things that might be dangerous to babies." The system responds by highlighting matching regions in the user's surroundings, as shown in Figure 5.7 Application 1, including unstable objects, potential falling hazards, and power outlets. Figure 5.9 illustrates more example queries and results.

This search capability is built on the language component of our scene volume, composed of a multi-channel CLIP feature volume. CLIP is key to this process as it embeds images and text into a shared feature space. Thus we can cast natural language search as building a map over the scene of the similarity between scene CLIP features and the CLIP embedding of a user-supplied text query. To enable this, we first resample the CLIP feature grid using trilinear interpolation to obtain a CLIP feature at each mesh vertex. We then compute the similarity of each vertex feature to the query feature, relative to a set of negative queries, following the search method of CLIP Surgery [205]. However, CLIP Surgery uses a long, fixed list of negative queries to identify redundant features that come at the cost of longer computation time and higher memory load, which exceed the acceptable limits for real-time interactive AR applications. We build the negative query list as the union of all class names extracted by our 3D semantic segmentation step. Our list is therefore shorter and more relevant, allowing us to produce heatmap outputs to the query similarity efficiently while filtering out noisy responses across the scene.

Leveraging this spatial search ability in AR applications can provide users with an

enhanced understanding and navigation of unfamiliar spaces. It can also enable them to explore complex environments faster than they would manually. Our Magic Leap 2 prototype application, as demonstrated in the accompanying video and Figure 5.7, overlays the response heatmap on top of the environment and physical objects via the optical-see-through display to provide an immersive user experience.

Intelligent Object Inventory

We imagine an intelligent AI AR companion that keeps a "temporal and spatial inventory" of objects for real-world environments, helping users keep track of the objects in their space. Integrating geometric and semantic knowledge into the joint 3D space makes it possible to automatically parse individual 3D objects from the environment for an object-centric user interface. As shown in our accompanying video and Figure 5.7 Day 1, the "magical" instantaneous highlighting and selection of physical objects in optical-see-through AR displays from any viewpoint provides an intuitive and direct interface for users to edit or personalize their space.

Our main goal with this application is to show that when coupled with in-situ learning, the multimodal-feature-fused environment models can unlock novel spatially-aware AI user interfaces. For instance, having access to intelligent virtual twins of every physical object in the room makes it feasible to train a machine-learning model to "learn and remember" physical objects, maintain object identities, and track object changes without aligning any noisy unstructured mesh models. To this end, we introduce a basic intelligent inventory system to visually present one interpretation of object changes in real-world environments.

We briefly discussed in Section 5.3.2 that the concept of "changes" from text-based version control systems does not automatically translate to the spatial, morphological, or appearance changes in physical objects. While the "true removal" of an object or

a paragraph of text is similar, naively comparing different scans of a space captured on different days yields counterproductive noise and does not maintain object identities. The ability to re-identify an arbitrary physical object is critical for effective user assistance. Learning to remember and re-identify an object relies on the in-situ learning model, which we introduced in Section 5.3.3. We will now discuss how it is used to reveal unchanged or missing objects in AR. This proof-of-concept AR demonstration does not yet constitute a full inventory system.

We offer three actions to collect user input to determine which objects to track and train on:

1) *Merge.* Users can merge multiple mesh segments into one if a single object was fragmented during the 3D reconstruction process, e.g., false boundaries introduced by shadows. This change is picked up by in-situ learning, leading to the recognition of segmented parts as the same object in future scans of this space.

2) *Rename.* Users can also customize the automatically generated object labels (e.g., "bottle:2" to "Joe's thermos") to improve their utility. This feature proves particularly beneficial in collaborative environments, such as a shared office, where it can help specify the ownership of items more clearly. In collaborative settings where multiple users might adjust the same space at different times, user-specified labels naturally reduce confusion. The accompanying video showcases an actual office setting in AR, where objects are distinctly tagged with their respective owner's names *(in the review version of the video, real names are blurred to maintain anonymity).*

3) *Remember.* Users can direct the system to track certain objects in the environment without further editing actions. The same objects should be re-identified with their current properties. This design provides a quick and easy way to collect "positive samples" that will be used to optimize the in-situ learning model for object classification.

The user's object-level personalization input provides the ground truth to guide the

learning objective. The in-situ learning model is trained to classify arbitrary objects based on their neural features from randomly sampled sparse graphs, which improves robustness across different scans and avoids overfitting. Specifically, objects that are merged, renamed, or remembered are flagged as "positive" samples with a unique label and assigned a class index in the classifier's ground truth. Through various design experiments, we arrived at a training strategy that classifies all other "non-personalized" objects as a "null" class with an index of zero. We sample null class features from all other voxels that do not belong to any of the personalized objects. While it performs robustly for re-identifying objects, this strategy limits the ability to differentiate newly introduced objects from the null objects, which we will discuss in Section 5.3.6. The user triggers the training after they finish personalization. The training stops automatically after the model reaches its peak accuracy (over 95% in our demon scene) plus certain cool-down epochs. Other strategies, such as adaptive learning rate, can also be used when users label new objects and fine-tune a trained model. In our office demo scene, we set the cool-down epochs to 10 and the total in-situ model training takes less than 8 seconds.

In pseudocode Figure 5.8, we describe how successive scans of a tracked space are processed and compared to analyze which objects have been removed or remained in the space. When a previously optimized in-situ model m is available, new semantically parsed voxel clusters (object segments or objects) from the new 3D scan F are first converted into a graph representation in data loader g and then sent to the in-situ model for classification, i.e., to check if it matches any object that was merged, renamed, or marked to remember by the user in previous scans ($m.labels$).

As we show in Figure 5.7 Application 2, scans from two different days of a tracked scene are akin to *"git commit"* states. The timestamps and chronology of the scans naturally form a version history. To easily inspect this history, our system renders a

```
# F: feature volumes with objects segmented
# m: in-situ model of a space
# m.labels: list of objects that m have trained on

previous_objs = {} or {Alice_desk, Bob_chair, ...}
current_objs = {}
unchanged_objs, missing_objs = [], []

for each object in F:
    # the initial scan of the space
    if not m.model_trained:
        # if, e.g., object is a desk
        current_objs["desk:1"] = object
        continue

    obj_graph = g(object) # voxels to graph
    obj_label = m(obj_graph) # in-situ prediction

    if not obj_label in m.labels: # unknown object
        current_objs["chair:6"] = object
        m.labels.append("chair:6")
    else: # object re-identified
        current_objs[obj_label] = object
        if obj_label in previous_objs:
            unchanged_objs.append(obj_label)
        else:
            missing_objs.append(obj_label)

# automatically labeled scene presented to users
# for personalization in AR, e.g.:
"desk:1" --> "Alice's desk"
"chair:6" --> "Bob's chair"

# train/finetune in-situ model
m.train_with(current_objs, m_labels)
m.model_trained = True
previous_objs = current_objs
```

Figure 5.8: Pseudocode describing how the Scene Manager and the user can build an intelligent object inventory with the in-situ learning model. After multimodal 3D fusion and post-processing, individual objects are passed through the in-situ model to re-identify previously existing objects and eventually reveal missing objects.

"volumetric diff", visualizing the different object inventory states over time:

1) *Unchanged objects.* Highlighting objects that were previously edited or marked by the user and can still be found in the current version of the environment. Additionally, objects that were previously merged or moved can still be recognized with their personalized names.

2) *Missing objects.* Revealing all objects that were present in the previous scan of the space but are now missing from the current version of the environment. Visualizing missing objects in red hollow contours in the current space lends users additional temporal awareness of their space. In the future, this new time dimension could be the substrate for novel and more complex AR interactions.

Critically, we produce the volumetric diff unlike any other methods discussed in Section 5.3.2. It is the automatically segmented objects and their vision-language features that are being compared by a neural network rather than versions of the reconstructed 3D model – misalignment in object or environment models will produce counterproductive noise instead of useful object tracking. In this proof-of-concept implementation, we manually aligned various scans to the physical room solely for the unchanged/missing objects' contours visualization. Re-identifying objects and listing which ones are unchanged or missing require no spatial alignment, which is not the focus of this work and is solvable through fiducial marker tracking algorithms (ARTag [206]) or model matching methods (3DMatch [207]).

5.3.5 In-situ learning and large language models

We introduced multimodal 3D fusion in Chapter 5 and showcased that fusing CLIP vision-language features into the environment and object models allows users to search in a physical space with natural-language queries. This language-infused 3D model has the

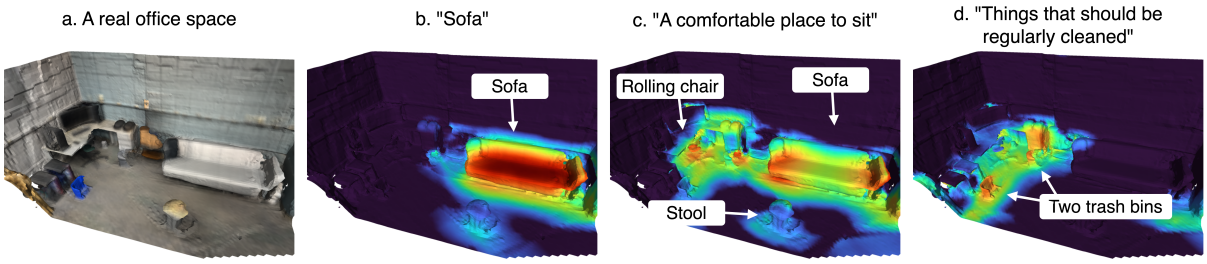


Figure 5.9: From object names to abstract natural-language queries, we show more examples of spatial search in a real-world environment.

obvious use case of quickly finding objects based on their description, but it can also be used creatively, leading to emergent utility, such as inspecting properties for damage or hazards, requesting decor advice, or mapping the layout of specific task-relevant items. Furthermore, this capability can serve as the backbone of future context-aware interaction systems, by identifying the virtual manipulations that are appropriate for each object in a flexible and open-ended way.

While working on the multimodal fusion pipeline and in-situ learning applications in AR 5.3.4, we experimented extensively with LLM APIs and local open-source LLMs. In this work, for the consideration of building a portable backpack system for real-time AR interactions, we decided against their use for the following reasons: a) The most capable LLMs remain cloud-based, making them problematic for privacy-sensitive AR applications; b) Although the throughput of LLM services is approaching real-time [208], the requirement of network traffic and added latency add friction to real-world AR use cases.

5.3.6 Limitations and Future Work

We implemented our in-situ learning for AR system with several deep-learning models that can run simultaneously on a single NVIDIA RTX 3090. The PC + Magic Leap 2 setup can easily be configured to a modern version of a portable backpack system such

as the "touring machine" [144]. Despite the encouraging results, these design choices: 1) constrained the linguistic understanding demonstration to spatial search in response to text queries instead of "actual conversations with physical spaces", and 2) limited the system's 3D object segmentation and object inventory feature to the 100 categories of common stuff and things defined in the COCO dataset [99, 209].

To achieve true open-vocabulary 3D object segmentation to support the tracking of any physical objects, adding the capability to recognize and train on any object not limited by pre-defined categories is straightforward. One viable solution is to adopt SegmentAnything [166] to identify object boundaries and then use LMMs such as LLaVA [172] or GPT-4V [171] for neural vision-language feature extraction and rich image description, labeling, or captioning.

Our current multimodal 3D scene model fusion approach opens new possibilities for context-aware AR interactions, such as responding to a natural-language query with a heatmap in AR (see Figure 5.9). Looking into the future of pervasive AR and human-AI teaming, we believe the ability to have a "back-and-forth conversation" with a physical space would be an attractive application for spatial computing. Think about asking your AR/AI system where in your backyard the best spots are to hang a hammock, or imagine you are a property manager and your conversational logging agent will proactively point out areas where closer inspection is needed based on previous findings spotted automatically during your current walk-through.

Conversing with LLMs in text (and even images) is now as easy as texting, thanks to open-source and commercial solutions [210, 171, 211]. However, unlike short conversations, conversing with physical spaces requires us to "tokenize the 3D model" and feed a large amount of "tokens" or feature representations into LMMs/LLMs to get a spatially meaningful response. This vision is theoretically possible with several tweaks based on our current implementation, yet there are several challenges worth considering:

- What is the best approach to "tokenize a $3m \times 3m$ space" that can maintain its spatial meaning and at the same time keep a good balance between accuracy (spatial and linguistic) and efficiency (fewer tokens)?

- Existing LLMs have a limited context window, the number of tokens a model can accept as the context for a response. Even with Google Gemini's seemingly large 1 million tokens window size [212], sending the entire 3D context to an LLM repeatedly is not yet a feasible approach.

- The throughput of LLM services (the number of output tokens per second) is approaching real-time for short contexts [208]. Yet, considering the number of tokens required for 3D contexts, it remains a challenging task for the real-time AR applications we pursue and demonstrated.

Our current intelligent object inventory implementation was designed to only recognize objects that are missing or unchanged in the physical world, but identifying "insertions" or previously unseen new objects could be a useful feature for real-world use cases. We discussed earlier that this limitation comes about because the training strategy is optimized for high accuracy for objects that appeared in previous 3D models. While we have seen more false positives than true positives in new object detection in ongoing experiments, inspirations from out-of-distribution detection [213] research could point us to potential solutions. Differentiating two or more identical objects poses a significant challenge for both humans and machines. By applying everyday human tricks, such as utilizing relative spatial relationships between objects, we may teach machines to enhance such capabilities. Our open-source implementation highlights areas that may benefit from future improved solutions and quantitative evaluations.

5.4 Implications of in-situ learning

In-situ machine learning empowers end users by placing them at the center of the AI-assisted system, including user inputs at various stages during the shaping of AI models – from user-led data collection that defines the AI's objective, to the user's direct evaluation of the AI's performance that determines the training process. This approach is particularly useful for non-expert users who can produce personalized AIs without the requirement for machine learning knowledge. The design of real-time collected data and continuously renewed user feedback also takes us a step closer to highly personalized AIs that adapt to the diverse preferences of different users.

In this chapter, we used three distinctive applications to showcase the real-world benefits of in-situ learning. The personalized pose detection system (Section 5.2.1) demonstrates the potential of flexible AI assistance that adapts to individuals' physical limitations and personal needs. The same system can be applied to other use cases beyond personal care. The segmentation tool that learns abstract vision or linguistic concepts (Section 5.2.2) not only reproduced a classic interactive machine learning example with more powerful capabilities, but also set the foundation for the intelligent object inventory system in our AR use case.

The new context-aware AR interfaces and intelligent object inventory system discussed in Section 5.3 were made possible by an in-situ learning AI system that trains on the spatially aware features from a custom multimodal 3D fusion pipeline that integrates neural vision-language features into the geometric and semantic representations of physical spaces. The intelligent inventory system re-identifies physical objects in AR and holds the promise to enhance personal space management, team collaboration and information exchange, and even asset management. The heightened temporal awareness of the physical spaces could potentially help tackle the issue of "change blindness" [214].

The ability to perform spatial searches using natural language within a 3D environment provides a glimpse into a more natural and intuitive AR interface – conversing with physical spaces for interior design suggestions, safety inspection visualizations, or personalized navigation.

We believe future human-AI teaming will feature highly personalized, rapidly optimized, and continuously improving AI teammates that understand the user and provide the right amount of assistance when needed in our daily lives. In-situ learning and our demonstrated use cases are our small contributions to this vision. We expect to see more potential for this human-AI system design philosophy in terms of applications and capabilities, particularly with the development of more powerful vision and vision language models that fundamentally enhance machines' sensing capabilities of our physical world.

Chapter 6

Summary and Discussion

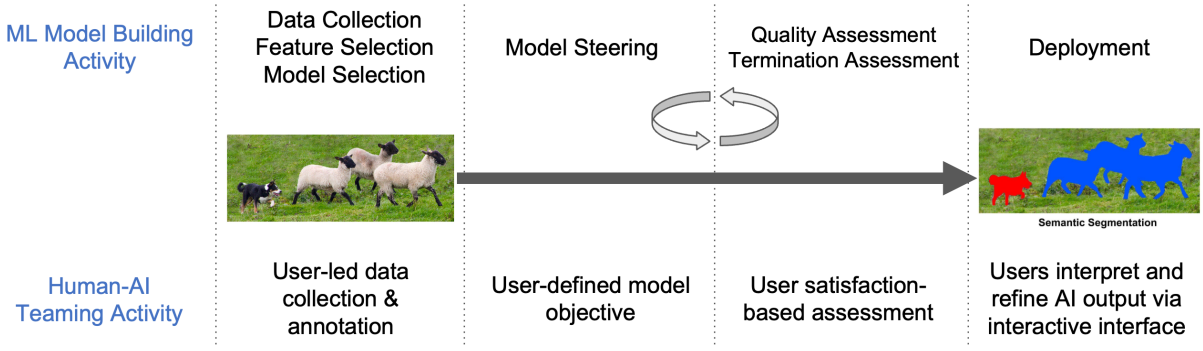


Figure 6.1: We identify the end-user activities that the various human-AI teaming projects have discussed and facilitated in this dissertation, corresponding to the ML model-building stages and activities.

6.1 Summary of contributions

One succinct summary of this dissertation's contribution is that we identified and facilitated various human-AI teaming opportunities for end users to participate and influence the model output and the shaping of the AI in real-world computer vision tasks. We discussed each chapter's contribution in terms of their application as well as their significance in facilitating human-AI teaming in Section 1.3. We also mapped each chapter

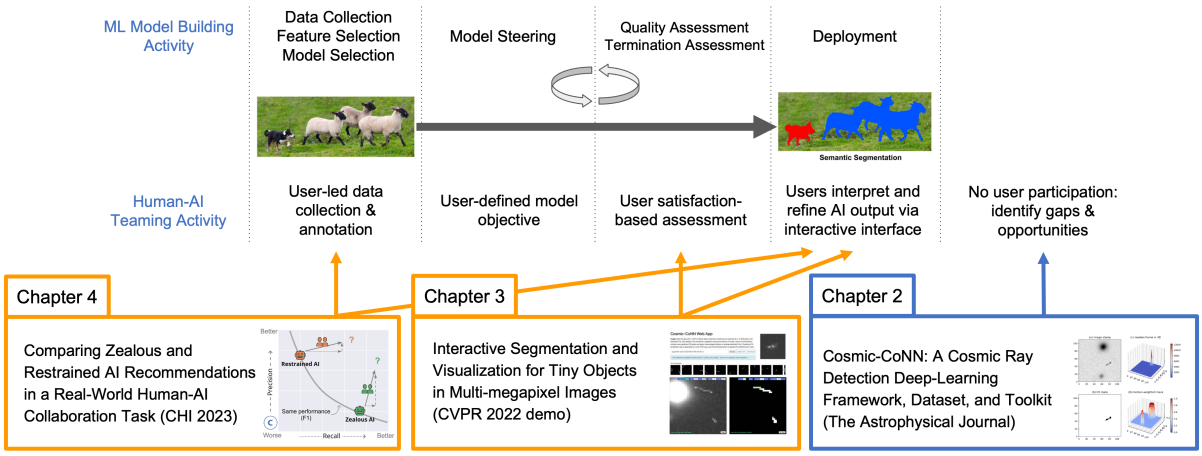


Figure 6.2: Mapping each chapter's contribution in terms of the end-user activities they facilitated during human-AI teaming.

to the AI system designer's activities in the model-building workflow [11].

Corresponding to these model machine learning building activities, in Figure 6.1, we further extend this workflow with human-AI teaming activities that are performed by end users. This complete overview from both the system designer's and end user's perspectives forms the basis for our summary discussion of these teaming opportunities. Let's revisit each chapter's significance in understanding and facilitating human-AI teaming in terms of end-user activities:

Chapter 2: Working closely with scientists highlights the critical role of their domain knowledge in applied machine learning, which helps identify the gaps between AI capabilities and user input in end-to-end models, pointing to a streamlined interface as an opportunity to interactively refine AI predictions.

Chapter 3: A streamlined interface can help democratize AI's capabilities and simplify the scientific imagery analysis workflow. The interactive model inference and editing allow users to better interpret and refine AI's prediction. The human-in-the-loop partnership enhances the black-box model's utility and the insights generated from the human-AI team. In this form of human-AI teaming, end users can evaluate the model's

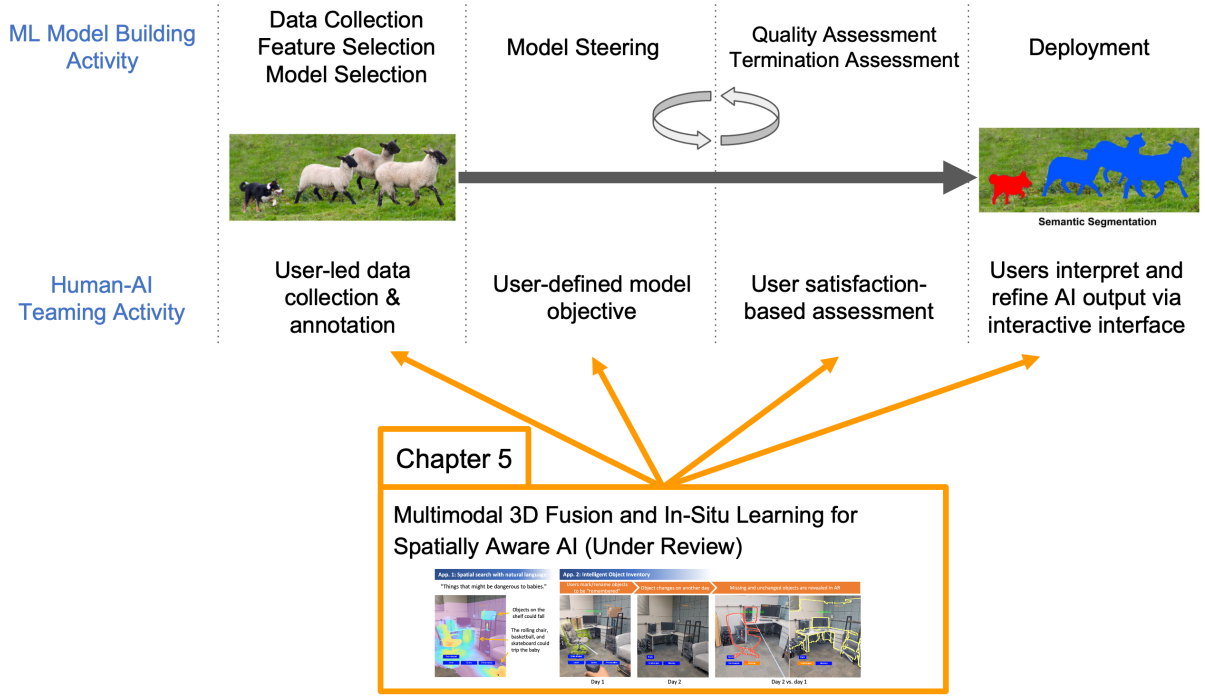


Figure 6.3: We propose the concept of in-situ learning in Chapter 5, which contributes to human-AI teaming as our proposed learning paradigm maximizes end user's participation and influence on the entire process during the shaping of ML models.

performance based on their domain knowledge, expertise, or personal subjective preferences. Such interfaces also paved the way for users to provide feedback for model improvement.

Chapter 4: An AI-assisted system carefully designed with an understanding of the task's priority and the inherent strengths and weaknesses of human collaborators can lead to improved team performance. On the contrary, naively partnering ill-suited AIs with humans can diminish human skills over time. Closely observing end users' behaviors, especially the unexpected ones, while working with an AI teammate can help AI system designers identify the optimizations to enhance the overall team performance and avoid hurting user skills (sustainable human-AI teaming).

Chapter 5: For real-world human-AI teaming, we propose in-situ learning to encode real-time user-collected data into a personalized neural network for knowledge storage

and real-world human-AI teaming assistance. The proposed learning paradigm is inspirational to future human-AI teaming, as it involves end users at every step of the model development process. Although this approach may currently be suitable for only a limited set of real-world applications, it significantly enhances the level of interaction between humans and AI, bringing them closer together than ever before.

6.2 Human-AI teaming for creative applications

Human-AI teaming is not only useful for tasks that require domain expertise or high-stakes decision making, as discussed earlier in this dissertation, it can also play a positive role in creative applications. Powerful text-to-image generative models like DALL·E [215] and Adobe Firefly[1] are now capable of generating high-quality images based on user's text descriptions of a scene. We demonstrate an example generated by Adobe Firefly in Figure 6.4. It is worth noting that human-AI teaming in text-to-image tasks creates opportunities for users to provide various forms of inputs to control or personalize the produced image, for example, by selecting specific areas for natural modification or using reference images that guide the expected look and layout of the output.

We show that by applying a purple cloud style reference image to the generated image in Figure 6.4, the user can achieve the expected purple sky effect. However, applying the reference style also altered many other components in the generated scene, such as the layout of objects, building size, material and the direction of the sun. This behavior highlights an important research topic: maintaining user control and accurately predicting user preference in generative models. Researchers often approach the problem with better user interface designs, enhanced visual-language modality alignment, etc.

[1]Adobe Firefly https://firefly.adobe.com

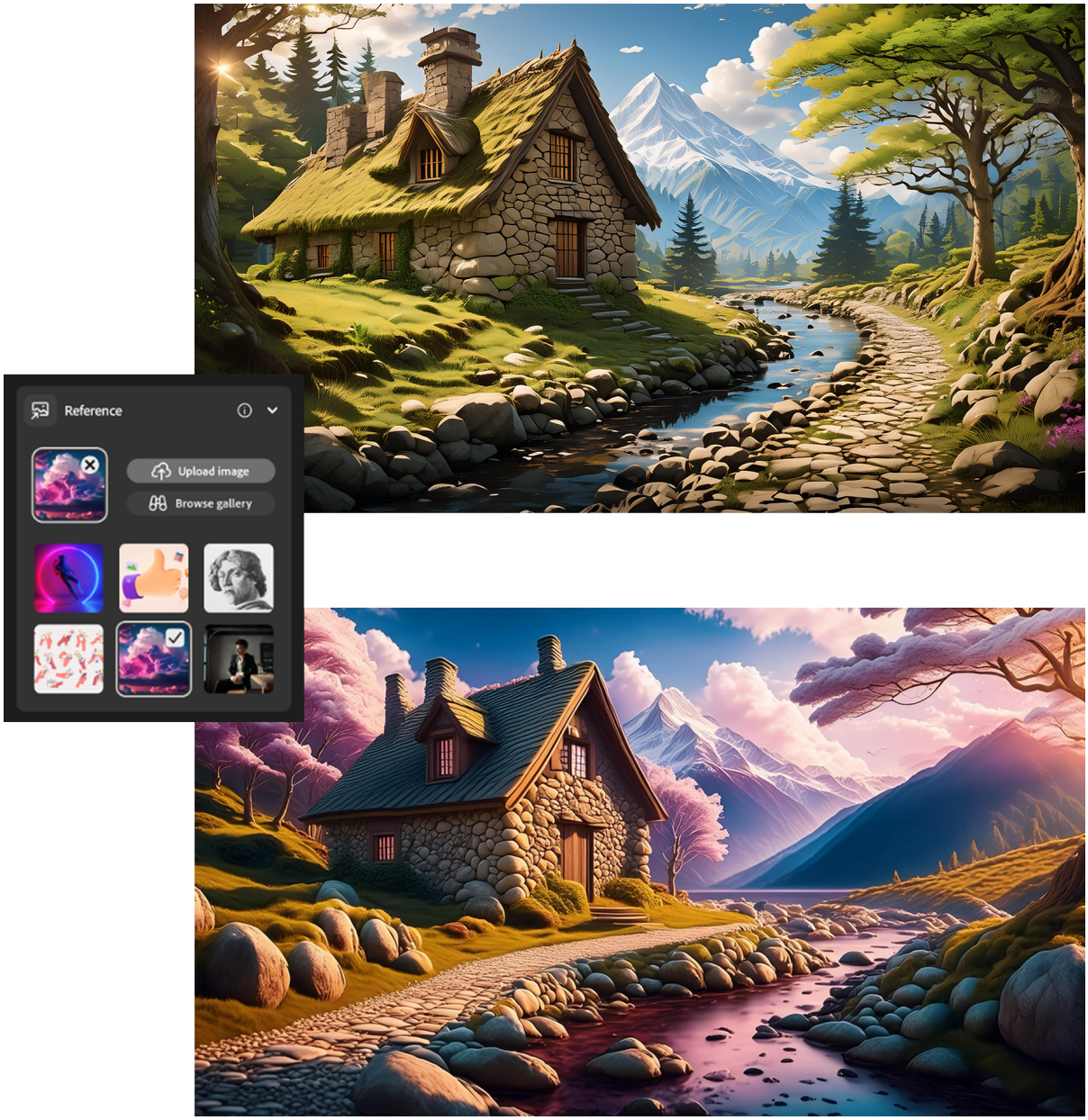


Figure 6.4: The example images are generated with Adobe Firefly Image 3 (preview) model (https://firefly.adobe.com) with the prompt of "Beautiful cozy fantasy stone cottage in a spring forest aside a cobblestone path and a babbling brook. Stone wall. Mountains in the distance. Magical tone and feel." Based on the first generated image, we applied a reference-style image of purple clouds (middle) to produce the second image.

We believe the interactive, iterative human-AI teaming approach that learns from the user, as described in in-situ learning (Chapter 5), and the lessons learned in this dissertation could provide a valid direction that helps artists maximize the utility of generative models while maintaining user control over the preferred elements and qualities. This is a future research direction in which the author of this dissertation is interested.

6.3 Human-AI teaming and the growing model size

In the context of interactive machine learning, it is critical to maintain the interactive "train-feedback-correct" cycles [14] to narrow the Gulf of Evaluation [216]. This also applies to human-AI teaming as timely feedback helps users learn the system's limitations and capabilities. Fiebrink et al. [217] made the observations that by evaluating results from the model, the users are developing effective strategies to build working systems – the system is also training the user to take appropriate actions to improve the joint performance.

Early interactive machine learning systems were able to steer the model on the fly with sparse input data from the user and provide instant feedback, thanks to the efficient and carefully chosen statistical learning algorithms for each of the specific use cases. Our literature research shows Support Vector Machines (SVM), Perceptron, Naive Bayes Classifiers, Decision Trees, and ensemble methods like AdaBoost, Random Forest, etc., are among the most frequently adopted algorithms in conventional IML applications. Many works also put strict constraints on the number of features for input data and model parameters to improve training efficiency and model transparency. Dimensionality reduction methods like principal components analysis (PCA), multidimensional scaling, and clustering are often used as ways to reduce input size, visualize model weights, or assist prediction interpretation. However, conventional statistical machine learning

methods are falling behind with the emergence of deep learning methods in complex computer vision tasks.

We witnessed the birth of deep neural networks, algorithms that formed the foundation of modern computer vision research, such as AlexNet [218], ZFNet [219], GoogLeNet [220], VGGNet [221], and ResNet [222] during the eight years of ImageNet Large Scale Visual Recognition Challenge (ILSVRC, 2010-17) [223]. Accelerated by the parallel computation on graphics processing units (GPUs), these deep learning models push the limit of ImageNet [224] classification task from 2011's $\sim 25\%$ top-5 error rate to less than 1% in less than ten years, exceeding human-level performance (5.1%) on the ImageNet dataset [225].

As computer vision models became more accurate, we also saw the growing size (number of trainable parameters) accompanying the improved model performance. Building on the achievements during the ImageNet era, computer vision research continues to evolve with the emergence of foundation models such as CLIP [143], DINO [226], SegmentAnything [166], and multi-modal language models like LLaVA [172] and GPT-4V [171]. These foundation models extend the capabilities of traditional vision systems, showcasing superior adaptability to diverse and complex scenes. However, they also further push the definition of large-scale models to the level of hundreds of millions of parameters to billions of parameters, making it impossible to finetune for real-time human-AI teaming.

In the age of billion-parameter models, it is now common to take many GPUs working in parallel for days, if not weeks, to optimize a large foundation model. If naively adopting modern large-scale deep neural networks in interactive machine learning or personalized human-AI teaming tasks, the prolonged training time will harm a user's effective evaluation of feedback provided or a change made to the system, i.e., a larger gap in the Gulf of Evaluation.

In response to the challenges posed by the increasing model size, we proposed in-

situ learning (Chapter 5) that utilizes compact features derived from foundation models to maintain the rapid "train-feedback-correct" cycles [14] even in complex vision tasks that required large-scale neural networks. By integrating these distilled features into smaller, task-specific models, in-situ learning allows for rapid model updates and adaptations based on user feedback or changing conditions without the overhead of extensive retraining. This method significantly accelerates the speed at which modifications can be evaluated and applied, effectively narrowing the Gulf of Evaluation, and making the system more responsive and efficient.

Nonetheless, with its own limitations like the requirements for real-time data collection and affordable user assessment for model quality evaluation, in-situ learning is not a universal solution for all human-AI teaming tasks. The increasing capabilities in foundation models and vision language models will continue to push AI system designers to tackle more complex tasks involving human-AI collaboration. Future research aimed at efficient feature extraction methods and effective model updates will be crucial for maximizing the potential of human-AI teaming systems.

Bibliography

[1] J. C. R. Licklider, *Man-computer symbiosis*, *IRE Transactions on Human Factors in Electronics* **HFE-1** (1960), no. 1 4–11.

[2] B. N. Patel, L. Rosenberg, G. Willcox, D. Baltaxe, M. Lyons, J. Irvin, P. Rajpurkar, T. Amrhein, R. Gupta, S. Halabi, *et. al.*, *Human–machine partnership with artificial intelligence for chest radiograph diagnosis*, *NPJ digital medicine* **2** (2019), no. 1 1–10.

[3] D. Wang, A. Khosla, R. Gargeya, H. Irshad, and A. H. Beck, *Deep learning for identifying metastatic breast cancer*, 2016.

[4] B. Brown, M. Broth, and E. Vinkhuyzen, *The halting problem: Video analysis of self-driving cars in traffic*, in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, (New York, NY, USA), Association for Computing Machinery, 2023.

[5] A. Lundgard, Y. Yang, M. L. Foster, and W. S. Lasecki, *Bolt: Instantaneous crowdsourcing via just-in-time training*, in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, (New York, NY, USA), p. 1–7, Association for Computing Machinery, 2018.

[6] E. Kamar, S. Hacker, and E. Horvitz, *Combining human and machine intelligence in large-scale crowdsourcing*, in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '12, (Richland, SC), p. 467–474, International Foundation for Autonomous Agents and Multiagent Systems, 2012.

[7] V. Lai, S. Carton, R. Bhatnagar, Q. V. Liao, Y. Zhang, and C. Tan, *Human-ai collaboration via conditional delegation: A case study of content moderation*, in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, (New York, NY, USA), Association for Computing Machinery, 2022.

[8] S. Feng and J. Boyd-Graber, *What can ai do for me? evaluating machine learning interpretations in cooperative play*, in *Proceedings of the 24th International*

Conference on Intelligent User Interfaces, IUI '19, (New York, NY, USA), p. 229–239, Association for Computing Machinery, 2019.

[9] Y. Nagar and T. W. Malone, *Making business predictions by combining human and machine intelligence in prediction markets*, in *International Conference on Information Systems*, Association for Information Systems, 2011.

[10] R. Tedrake, *Robotic Manipulation*. 2023.

[11] J. J. Dudley and P. O. Kristensson, *A review of user interface design for interactive machine learning*, *ACM Trans. Interact. Intell. Syst.* **8** (jun, 2018).

[12] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, *Power to the people: The role of humans in interactive machine learning*, *AI Magazine* **35** (Dec., 2014) 105–120.

[13] M. Ware, E. FRANK, G. HOLMES, M. HALL, and I. H. WITTEN, *Interactive machine learning: letting users build classifiers*, *International Journal of Human-Computer Studies* **55** (2001), no. 3 281–292.

[14] J. A. Fails and D. R. Olsen, *Interactive machine learning*, in *Proceedings of the 8th International Conference on Intelligent User Interfaces*, IUI '03, (New York, NY, USA), p. 39–45, Association for Computing Machinery, 2003.

[15] C. Xu, C. McCully, B. Dong, D. A. Howell, and P. Sen, *Cosmic-CoNN: A cosmic-ray detection deep-learning framework, data set, and toolkit*, . Publisher: The American Astronomical Society.

[16] N. D. Miles, S. E. Deustua, G. Tancredi, G. Schnyder, S. Nesmachnow, and G. Cromwell, *Using cosmic rays detected by hst as geophysical markers. i. detection and characterization of cosmic rays*, *The Astrophysical Journal* **918** (sep, 2021) 86.

[17] R. A. Windhorst, B. E. Franklin, and L. W. Neuschaefer, *Removing Cosmic-Ray Hits from Multi-Orbit HST Wide Field Camera Images*, *ublications of the Astronomical Society of the Pacific* **106** (July, 1994) 798.

[18] C. Y. Zhang, *Robust Estimation and Image Combining*, in *Astronomical Data Analysis Software and Systems IV* (R. A. Shaw, H. E. Payne, and J. J. E. Hayes, eds.), vol. 77 of *Astronomical Society of the Pacific Conference Series*, p. 514, Jan., 1995.

[19] W. Freudling, *An Image-Restoration Technique for the Removal of Cosmic Ray Hits from Dithered Images*, *ublications of the Astronomical Society of the Pacific* **107** (Jan., 1995) 85.

[20] A. S. Fruchter and R. N. Hook, *Drizzle: A Method for the Linear Reconstruction of Undersampled Images*, *ublications of the Astronomical Society of the Pacific* **114** (Feb., 2002) 144–152, [astro-ph/9808087].

[21] S. Desai, J. J. Mohr, E. Bertin, M. Kümmel, and M. Wetzstein, *Detection and removal of artifacts in astronomical images*, *Astronomy and Computing* **16** (July, 2016) 67–78, [arXiv:1601.0718].

[22] J. E. Rhoads, *Cosmic-Ray Rejection by Linear Filtering of Single Images*, *ublications of the Astronomical Society of the Pacific* **112** (May, 2000) 703–710, [astro-ph/0002041].

[23] W. Pych, *A Fast Algorithm for Cosmic-Ray Removal from Single Images*, *ublications of the Astronomical Society of the Pacific* **116** (Feb., 2004) 148–153, [astro-ph/0311290].

[24] L. Shamir, *A fuzzy logic-based algorithm for cosmic-ray hit rejection from single images*, *Astronomische Nachrichten* **326** (July, 2005) 428–431.

[25] P. G. van Dokkum, *Cosmic-Ray Rejection by Laplacian Edge Detection*, *ublications of the Astronomical Society of the Pacific* **113** (Nov., 2001) 1420–1427, [astro-ph/0108003].

[26] F. D. Murtagh and H. M. Adorf, *Detecting Cosmic-Ray Hits on HST Wf/pc Images*, in *European Southern Observatory Conference and Workshop Proceedings*, vol. 38 of *European Southern Observatory Conference and Workshop Proceedings*, p. 51, Jan., 1991.

[27] S. Salzberg, R. Chandar, H. Ford, S. K. Murthy, and R. White, *Decision Trees for Automated Identification of Cosmic-Ray Hits in Hubble Space Telescope Images*, *ublications of the Astronomical Society of the Pacific* **107** (Mar., 1995) 279.

[28] D. Baron, *Machine Learning in Astronomy: a practical overview*, *arXiv e-prints* (Apr., 2019) arXiv:1904.07248, [arXiv:1904.0724].

[29] K. Zhang and J. S. Bloom, *deepCR: Cosmic Ray Rejection with Deep Learning*, *Astrophysical Journal* **889** (Jan., 2020) 24, [arXiv:1907.0950].

[30] J. S. Chen, A. Huertas, and G. Medioni, *Fast convolution with laplacian-of-gaussian masks*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-9** (1987), no. 4 584–590.

[31] O. Ronneberger, P. Fischer, and T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, *arXiv e-prints* (May, 2015) arXiv:1505.04597, [arXiv:1505.0459].

[32] M. Buda, A. Maki, and M. A. Mazurowski, *A systematic study of the class imbalance problem in convolutional neural networks*, *Neural Networks* **106** (2018) 249 – 259.

[33] C. McCully, N. H. Volgenau, D.-R. Harbeck, T. A. Lister, E. S. Saunders, M. L. Turner, R. J. Siiverd, and M. Bowman, *Real-time processing of the imaging data from the network of Las Cumbres Observatory Telescopes using BANZAI*, in *Software and Cyberinfrastructure for Astronomy V* (J. C. Guzman and J. Ibsen, eds.), vol. 10707 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, p. 107070K, July, 2018. arXiv:1811.0416.

[34] C. L. Farage and K. A. Pimbblet, *Evaluation of Cosmic Ray Rejection Algorithms on Single-Shot Exposures*, *Publications of the Astron. Soc. of Australia* **22** (Aug., 2005) 249–256, [astro-ph/0506476].

[35] W. Little, *The existence of persistent states in the brain*, *Mathematical Biosciences* **19** (1974), no. 1 101–120.

[36] W. Little and G. L. Shaw, *Analytic study of the memory storage capacity of a neural network*, *Mathematical Biosciences* **39** (1978), no. 3 281–290.

[37] T. J. Sørensen, *A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons*, vol. 5. Munksgaard Copenhagen, 1948.

[38] J. Kiefer and J. Wolfowitz, *Stochastic estimation of the maximum of a regression function*, *The Annals of Mathematical Statistics* (1952) 462–466.

[39] L. Bottou, F. E. Curtis, and J. Nocedal, *Optimization Methods for Large-Scale Machine Learning*, *arXiv e-prints* (June, 2016) arXiv:1606.04838, [arXiv:1606.0483].

[40] S. Ioffe and C. Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015* (F. R. Bach and D. M. Blei, eds.), vol. 37 of *JMLR Workshop and Conference Proceedings*, pp. 448–456, JMLR.org, 2015.

[41] Y. Wu and K. He, *Group normalization*, in *Proceedings of the European Conference on Computer Vision (ECCV)*, September, 2018.

[42] R. E. González, R. P. Muñoz, and C. A. Hernández, *Galaxy detection and identification using deep learning and data augmentation*, *Astronomy and Computing* **25** (Oct., 2018) 103–109, [arXiv:1809.0169].

[43] F. C. Gillett, M. Mountain, R. Kurz, D. A. Simons, M. G. Smith, and T. Boroson, *The Gemini Telescopes Project (Invited Paper)*, in *Revista Mexicana de Astronomia y Astrofisica Conference Series* (E. Falco, J. A. Fernandez, and R. F. Ferrero, eds.), vol. 4 of *Revista Mexicana de Astronomia y Astrofisica Conference Series*, p. 75, Nov., 1996.

[44] T. Fawcett, *An introduction to ROC analysis*, *Pattern Recognit. Lett.* **27** (2006), no. 8 861–874.

[45] T. Saito and M. Rehmsmeier, *The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets*, *PLoS ONE* **10** (Mar., 2015) e0118432.

[46] D. Hiramatsu, D. A. Howell, S. D. Van Dyk, J. A. Goldberg, K. Maeda, T. J. Moriya, N. Tominaga, K. Nomoto, G. Hosseinzadeh, I. Arcavi, C. McCully, J. Burke, K. A. Bostroem, S. Valenti, Y. Dong, P. J. Brown, J. E. Andrews, C. Bilinski, G. G. Williams, P. S. Smith, N. Smith, D. J. Sand, G. S. Anand, C. Xu, A. V. Filippenko, M. C. Bersten, G. Folatelli, P. L. Kelly, T. Noguchi, and K. Itagaki, *The electron-capture origin of supernova 2018zd*, *Nature Astronomy* **5** (June, 2021) 903–910, [arXiv:2011.0217].

[47] C. McCully, S. Crawford, G. Kovacs, E. Tollerud, E. Betts, L. Bradley, M. Craig, J. Turner, O. Streicher, B. Sipocz, T. Robitaille, and C. Deil, *astropy/astroscrappy: v1.0.5 zenodo release*, Nov., 2018.

[48] K. Labrie, K. Anderson, R. Cárdenes, C. Simpson, and J. E. H. Turner, *DRAGONS - Data Reduction for Astronomy from Gemini Observatory North and South*, in *Astronomical Data Analysis Software and Systems XXVII* (P. J. Teuben, M. W. Pound, B. A. Thomas, and E. M. Warner, eds.), vol. 523 of *Astronomical Society of the Pacific Conference Series*, p. 321, Oct., 2019.

[49] S. Bhavanam, S. Channappayya, P. Srijith, and S. Desai, *Cosmic ray rejection with attention augmented deep learning*, *Astronomy and Computing* **40** (2022) 100625.

[50] B. Flaugher, H. T. Diehl, K. Honscheid, T. M. C. Abbott, O. Alvarez, R. Angstadt, J. T. Annis, M. Antonik, O. Ballester, L. Beaufore, G. M. Bernstein, R. A. Bernstein, B. Bigelow, M. Bonati, D. Boprie, D. Brooks, E. J. Buckley-Geer, J. Campa, L. Cardiel-Sas, F. J. Castander, J. Castilla, H. Cease, J. M. Cela-Ruiz, S. Chappa, E. Chi, C. Cooper, L. N. da Costa, E. Dede, G. Derylo, D. L. DePoy, J. de Vicente, P. Doel, A. Drlica-Wagner, J. Eiting, A. E. Elliott, J. Emes, J. Estrada, A. F. Neto, D. A. Finley, R. Flores, J. Frieman, D. Gerdes, M. D. Gladders, B. Gregory, G. R. Gutierrez, J. Hao, S. E. Holland, S. Holm, D. Huffman, C. Jackson, D. J. James, M. Jonas, A. Karcher, I. Karliner, S. Kent, R. Kessler, M. Kozlovsky, R. G. Kron, D. Kubik, K. Kuehn,

S. Kuhlmann, K. Kuk, O. Lahav, A. Lathrop, J. Lee, M. E. Levi, P. Lewis, T. S. Li, I. Mandrichenko, J. L. Marshall, G. Martinez, K. W. Merritt, R. Miquel, F. Muñoz, E. H. Neilsen, R. C. Nichol, B. Nord, R. Ogando, J. Olsen, N. Palaio, K. Patton, J. Peoples, A. A. Plazas, J. Rauch, K. Reil, J.-P. Rheault, N. A. Roe, H. Rogers, A. Roodman, E. Sanchez, V. Scarpine, R. H. Schindler, R. Schmidt, R. Schmitt, M. Schubnell, K. Schultz, P. Schurter, L. Scott, S. Serrano, T. M. Shaw, R. C. Smith, M. Soares-Santos, A. Stefanik, W. Stuermer, E. Suchyta, A. Sypniewski, G. Tarle, J. Thaler, R. Tighe, C. Tran, D. Tucker, A. R. Walker, G. Wang, M. Watson, C. Weaverdyck, W. Wester, R. Woods, and B. Y. and, *THE DARK ENERGY CAMERA*, *The Astronomical Journal* **150** (oct, 2015) 150.

[51] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, and D. Rueckert, *Attention u-net: Learning where to look for the pancreas*, in *Medical Imaging with Deep Learning*, 2018.

[52] K. J. Kwon, K. Zhang, and J. S. Bloom, *deepCR on ACS/WFC: Cosmic-ray rejection for HST ACS/WFC photometry*, *Research Notes of the AAS* **5** (apr, 2021) 98.

[53] Z. Bai, H. Zhang, H. Yuan, J. L. Carlin, G. Li, Y. Lei, Y. Dong, H. Yang, Y. Zhao, and Z. Cao, *Cosmic Ray Removal in Fiber Spectroscopic Image*, *ublications of the Astronomical Society of the Pacific* **129** (Feb., 2017) 024004, [arXiv:1705.0208].

[54] C. Xu, B. Dong, N. Stier, C. McCully, D. A. Howell, P. Sen, and T. Höllerer, *Interactive segmentation and visualization for tiny objects in multi-megapixel images*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21447–21452, June, 2022.

[55] W. A. Joye and E. Mandel, *New Features of SAOImage DS9*, in *Astronomical Data Analysis Software and Systems XII* (H. E. Payne, R. I. Jedrzejewski, and R. N. Hook, eds.), vol. 295 of *Astronomical Society of the Pacific Conference Series*, p. 489, Jan., 2003.

[56] Ž. Ivezić, S. M. Kahn, J. A. Tyson, B. Abel, E. Acosta, R. Allsman, D. Alonso, Y. AlSayyad, S. F. Anderson, J. Andrew, and et al., *LSST: From Science Drivers to Reference Design and Anticipated Data Products*, *Astrophysical Journal* **873** (Mar., 2019) 111, [arXiv:0805.2366].

[57] Astropy Collaboration, T. P. Robitaille, E. J. Tollerud, P. Greenfield, M. Droettboom, E. Bray, T. Aldcroft, M. Davis, A. Ginsburg, A. M. Price-Whelan, W. E. Kerzendorf, A. Conley, N. Crighton, K. Barbary, D. Muna, H. Ferguson, F. Grollier, M. M. Parikh, P. H. Nair, H. M. Unther, C. Deil, J. Woillez, S. Conseil, R. Kramer, J. E. H. Turner, L. Singer, R. Fox, B. A. Weaver, V. Zabalza, Z. I. Edwards, K. Azalee Bostroem, D. J. Burke, A. R.

Casey, S. M. Crawford, N. Dencheva, J. Ely, T. Jenness, K. Labrie, P. L. Lim, F. Pierfederici, A. Pontzen, A. Ptak, B. Refsdal, M. Servillat, and O. Streicher, *Astropy: A community Python package for astronomy*, *Astronomy and Astrophysics* **558** (Oct., 2013) A33, [arXiv:1307.6212].

[58] Astropy Collaboration, A. M. Price-Whelan, B. M. Sipőcz, H. M. Günther, P. L. Lim, S. M. Crawford, S. Conseil, D. L. Shupe, M. W. Craig, N. Dencheva, A. Ginsburg, J. T. Vand erPlas, L. D. Bradley, D. Pérez-Suárez, M. de Val-Borro, T. L. Aldcroft, K. L. Cruz, T. P. Robitaille, E. J. Tollerud, C. Ardelean, T. Babej, Y. P. Bach, M. Bachetti, A. V. Bakanov, S. P. Bamford, G. Barentsen, P. Barmby, A. Baumbach, K. L. Berry, F. Biscani, M. Boquien, K. A. Bostroem, L. G. Bouma, G. B. Brammer, E. M. Bray, H. Breytenbach, H. Buddelmeijer, D. J. Burke, G. Calderone, J. L. Cano Rodríguez, M. Cara, J. V. M. Cardoso, S. Cheedella, Y. Copin, L. Corrales, D. Crichton, D. D'Avella, C. Deil, É. Depagne, J. P. Dietrich, A. Donath, M. Droettboom, N. Earl, T. Erben, S. Fabbro, L. A. Ferreira, T. Finethy, R. T. Fox, L. H. Garrison, S. L. J. Gibbons, D. A. Goldstein, R. Gommers, J. P. Greco, P. Greenfield, A. M. Groener, F. Grollier, A. Hagen, P. Hirst, D. Homeier, A. J. Horton, G. Hosseinzadeh, L. Hu, J. S. Hunkeler, Ž. Ivezić, A. Jain, T. Jenness, G. Kanarek, S. Kendrew, N. S. Kern, W. E. Kerzendorf, A. Khvalko, J. King, D. Kirkby, A. M. Kulkarni, A. Kumar, A. Lee, D. Lenz, S. P. Littlefair, Z. Ma, D. M. Macleod, M. Mastropietro, C. McCully, S. Montagnac, B. M. Morris, M. Mueller, S. J. Mumford, D. Muna, N. A. Murphy, S. Nelson, G. H. Nguyen, J. P. Ninan, M. Nöthe, S. Ogaz, S. Oh, J. K. Parejko, N. Parley, S. Pascual, R. Patil, A. A. Patil, A. L. Plunkett, J. X. Prochaska, T. Rastogi, V. Reddy Janga, J. Sabater, P. Sakurikar, M. Seifert, L. E. Sherbert, H. Sherwood-Taylor, A. Y. Shih, J. Sick, M. T. Silbiger, S. Singanamalla, L. P. Singer, P. H. Sladen, K. A. Sooley, S. Sornarajah, O. Streicher, P. Teuben, S. W. Thomas, G. R. Tremblay, J. E. H. Turner, V. Terrón, M. H. van Kerkwijk, A. de la Vega, L. L. Watkins, B. A. Weaver, J. B. Whitmore, J. Woillez, V. Zabalza, and Astropy Contributors, *The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package*, *Astronomical Journal* **156** (Sept., 2018) 123, [arXiv:1801.0263].

[59] C. Xu and boningdong, *cy-xu/cosmic-conn: v0.4.1*, June, 2022.

[60] C. Xu, C. McCully, B. Dong, D. A. Howell, and P. Sen, "Cosmic-CoNN: Cosmic ray detection toolkit." Astrophysics Source Code Library, record ascl:2108.018, Aug., 2021.

[61] T. Robitaille, C. Deil, and A. Ginsburg, *reproject: Python-based astronomical image reprojection*, Nov., 2020.

[62] J. D. Hunter, *Matplotlib: A 2d graphics environment*, *Computing in Science & Engineering* **9** (2007), no. 3 90–95.

[63] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del R'ıo, M. Wiebe, P. Peterson, P. G'erard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, *Array programming with NumPy*, *Nature* **585** (Sept., 2020) 357–362.

[64] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors, *scikit-image: image processing in Python*, *PeerJ* **2** (6, 2014) e453.

[65] E. Bertin and S. Arnouts, *SExtractor: Software for source extraction.*, *Astronomy and Astrophysicss* **117** (June, 1996) 393–404.

[66] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, *Pytorch: An imperative style, high-performance deep learning library*, in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.

[67] T. M. Brown, N. Baliber, F. B. Bianco, M. Bowman, B. Burleson, P. Conway, M. Crellin, É. Depagne, J. D. Vera, B. Dilday, D. Dragomir, M. Dubberley, J. D. Eastman, M. Elphick, M. Falarski, S. Foale, M. Ford, B. J. Fulton, J. Garza, E. L. Gomez, M. Graham, R. Greene, B. Haldeman, E. Hawkins, B. Haworth, R. Haynes, M. Hidas, A. E. Hjelstrom, D. A. Howell, J. Hygelund, T. A. Lister, R. Lobdill, J. Martinez, D. S. Mullins, M. Norbury, J. Parrent, R. Paulson, D. L. Petry, A. Pickles, V. Posner, W. E. Rosing, R. Ross, D. J. Sand, E. S. Saunders, J. Shobbrook, A. Shporer, R. A. Street, D. Thomas, Y. Tsapras, J. R. Tufts, S. Valenti, K. V. Horst, Z. Walker, G. White, and M. Willis, *Las cumbres observatory global telescope network*, *Publications of the Astronomical Society of the Pacific* **125** (sep, 2013) 1031–1055.

[68] T. Robitaille, A. Ginsburg, and C. Deil, *astropy/reproject*, 2019.

[69] K. Barbary, *Sep: Source extractor as a library*, *Journal of Open Source Software* **1** (2016), no. 6 58.

[70] S. B. Howell, *Photometry and astrometry*, p. 102–134. Cambridge Observing Handbooks for Research Astronomers. Cambridge University Press, 2 ed., 2006.

[71] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, *arXiv e-prints* (Dec., 2014) arXiv:1412.6980, [arXiv:1412.6980].

[72] R. A. Windhorst, B. E. Franklin, and L. W. Neuschaefer, *Removing cosmic-ray hits from multiorbit HST wide field camera images*, *Publications of the Astronomical Society of the Pacific* **106** (jul, 1994) 798.

[73] P. G. van Dokkum, *Cosmic-ray rejection by laplacian edge detection*, *Publications of the Astronomical Society of the Pacific* **113** (nov, 2001) 1420–1427.

[74] D. Groom, *Cosmic rays and other nonsense in astronomical ccd imagers*, in *Scientific Detectors for Astronomy* (P. Amico, J. W. Beletic, and J. E. Beletic, eds.), (Dordrecht), pp. 81–94, Springer Netherlands, 2004.

[75] S. M. Kahn, N. Kurita, K. Gilmore, M. Nordby, P. O'Connor, R. Schindler, J. Oliver, R. V. Berg, S. Olivier, V. Riot, P. Antilogus, T. Schalk, M. Huffer, G. Bowden, J. Singal, and M. Foss, *Design and development of the 3.2 gigapixel camera for the Large Synoptic Survey Telescope*, in *Ground-based and Airborne Instrumentation for Astronomy III* (I. S. McLean, S. K. Ramsay, and H. Takami, eds.), vol. 7735, pp. 257 – 273, International Society for Optics and Photonics, SPIE, 2010.

[76] C. Xu, C. Mccully, B. Dong, D. A. Howell, and P. Sen, *Cosmic-conn: A cosmic ray detection deep-learning framework, dataset, and toolkit*, 2021.

[77] C. Xu, C. McCully, B. Dong, D. A. Howell, and P. Sen, *Cosmic-conn lco cr dataset*, June, 2021.

[78] C. McCully, M. Turner, N. Volgenau, D. Harbeck, S. Valenti, A. Riba, E. Bachelet, I. W. Snyder, B. Kurczynski, M. Norbury, and R. Street, *Lcogt/banzai: Initial release*, June, 2018.

[79] W. A. Joye and E. Mandel, *The Development of SAOImage DS9: Lessons Learned from a Small but Successful Software Project*, in *Astronomical Data Analysis Software and Systems XIV* (P. Shopbell, M. Britton, and R. Ebert, eds.), vol. 347 of *Astronomical Society of the Pacific Conference Series*, p. 110, Dec., 2005.

[80] C. A. Schneider, W. S. Rasband, and K. W. Eliceiri, *Nih image to imagej: 25 years of image analysis*, 2012.

[81] C. T. Rueden, J. Schindelin, M. C. Hiner, B. E. DeZonia, A. E. Walter, E. T. Arena, and K. W. Eliceiri, *Imagej2: Imagej for the next generation of scientific image data*, *BMC Bioinformatics* **18** (11, 2017) 1–26.

[82] K. A. Collins, J. F. Kielkopf, K. G. Stassun, and F. V. Hessman, *Astroimagej: image processing and photometric extraction for ultra-precise astronomical light curves*, *The Astronomical Journal* **153** (2017), no. 2 77.

[83] E. Gómez-de Mariscal, C. García-López-de Haro, W. Ouyang, L. Donati, E. Lundberg, M. Unser, A. Muñoz-Barrutia, and D. Sage, *Deepimagej: A user-friendly environment to run deep learning models in imagej*, *Nature Methods* **18** (2021), no. 10 1192–1195.

[84] P. A. Yushkevich, J. Piven, H. Cody Hazlett, R. Gimpel Smith, S. Ho, J. C. Gee, and G. Gerig, *User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability*, *Neuroimage* **31** (2006), no. 3 1116–1128.

[85] A. Bussone, S. Stumpf, and D. O'Sullivan, *The role of explanations on trust and reliance in clinical decision support systems*, in *2015 International Conference on Healthcare Informatics*, pp. 160–169, 2015.

[86] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, *Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission*, in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, (New York, NY, USA), p. 1721–1730, Association for Computing Machinery, 2015.

[87] Y. Xie, M. Chen, D. Kao, G. Gao, and X. A. Chen, *Chexplain: Enabling physicians to explore and understand data-driven, ai-enabled medical imaging analysis*, in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, (New York, NY, USA), p. 1–13, Association for Computing Machinery, 2020.

[88] N. Van Berkel, J. Opie, O. F. Ahmad, L. Lovat, D. Stoyanov, and A. Blandford, *Initial responses to false positives in ai-supported continuous interactions: A colonoscopy case study*, *ACM Trans. Interact. Intell. Syst.* **12** (mar, 2022).

[89] M. Colley, B. Eder, J. O. Rixen, and E. Rukzio, *Effects of semantic segmentation visualization on trust, situation awareness, and cognitive load in highly automated vehicles*, in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, (New York, NY, USA), Association for Computing Machinery, 2021.

[90] Y. Chen, X. Zhang, and J. Wang, *Robust vehicle driver assistance control for handover scenarios considering driving performances*, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **51** (2021), no. 7 4160–4170.

[91] D. Gopinath, J. DeCastro, G. Rosman, E. Sumner, A. Morgan, S. Hakimi, and S. Stent, *Hmiway-env: A framework for simulating behaviors and preferences to support human-ai teaming in driving*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 4342–4350, June, 2022.

[92] G. Bansal, B. Nushi, E. Kamar, E. Horvitz, and D. S. Weld, *Is the most accurate ai the best teammate? optimizing ai for teamwork*, *Proceedings of the AAAI Conference on Artificial Intelligence* **35** (May, 2021) 11405–11414.

[93] G. Bansal, T. Wu, J. Zhou, R. Fok, B. Nushi, E. Kamar, M. T. Ribeiro, and D. Weld, *Does the whole exceed its parts? the effect of ai explanations on complementary team performance*, in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, (New York, NY, USA), Association for Computing Machinery, 2021.

[94] Q. Zhang, M. L. Lee, and S. Carter, *You complete me: Human-ai teams and complementary expertise*, in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, (New York, NY, USA), Association for Computing Machinery, 2022.

[95] Y. Sasaki *et. al.*, *The truth of the f-measure*, *Teach tutor mater* **1** (2007), no. 5 1–5.

[96] G. Csurka, D. Larlus, and F. Perronnin, *What is a good evaluation measure for semantic segmentation?*, in *Procedings of the British Machine Vision Conference 2013*, (Bristol), pp. 32.1–32.11, British Machine Vision Association, 2013.

[97] J. Davis and M. Goadrich, *The relationship between Precision-Recall and ROC curves*, in *Proceedings of the 23rd international conference on Machine learning*, ICML '06, (New York, NY, USA), pp. 233–240, Association for Computing Machinery, June, 2006.

[98] Z. C. Lipton, C. Elkan, and B. Naryanaswamy, *Optimal Thresholding of Classifiers to Maximize F1 Measure*, in *Machine Learning and Knowledge Discovery in Databases* (T. Calders, F. Esposito, E. Hüllermeier, and R. Meo, eds.), Lecture Notes in Computer Science, (Berlin, Heidelberg), pp. 225–239, Springer, 2014.

[99] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, *Microsoft COCO: Common Objects in Context*, in *Computer Vision – ECCV 2014*, Lecture Notes in Computer Science, (Cham), pp. 740–755, Springer International Publishing, 2014.

[100] Z. Erenel and H. Altınçay, *Improving the precision-recall trade-off in undersampling-based binary text categorization using unanimity rule*, *Neural Computing and Applications* **22** (May, 2013) 83–100.

[101] F. Morstatter, L. Wu, T. H. Nazer, K. M. Carley, and H. Liu, *A new approach to bot detection: Striking the balance between precision and recall*, in *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 533–540, 2016.

[102] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, M. Martin, T. Nagarajan, I. Radosavovic, S. K. Ramakrishnan, F. Ryan, J. Sharma, M. Wray, M. Xu, E. Z. Xu, C. Zhao, S. Bansal, D. Batra, V. Cartillier, S. Crane, T. Do, M. Doulaty, A. Erapalli, C. Feichtenhofer, A. Fragomeni, Q. Fu, C. Fuegen, A. Gebreselasie, C. Gonzalez, J. Hillis, X. Huang, Y. Huang, W. Jia, W. Khoo, J. Kolar, S. Kottur, A. Kumar, F. Landini, C. Li, Y. Li, Z. Li, K. Mangalam, R. Modhugu, J. Munro, T. Murrell, T. Nishiyasu, W. Price, P. R. Puentes, M. Ramazanova, L. Sari, K. Somasundaram, A. Southerland, Y. Sugano, R. Tao, M. Vo, Y. Wang, X. Wu, T. Yagi, Y. Zhu, P. Arbelaez, D. Crandall, D. Damen, G. M. Farinella, B. Ghanem, V. K. Ithapu, C. V. Jawahar, H. Joo, K. Kitani, H. Li, R. Newcombe, A. Oliva, H. S. Park, J. M. Rehg, Y. Sato, J. Shi, M. Z. Shou, A. Torralba, L. Torresani, M. Yan, and J. Malik, *Ego4d: Around the World in 3,000 Hours of Egocentric Video*, in *IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2022.

[103] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, *Bytetrack: Multi-object tracking by associating every detection box*, .

[104] G. Bansal, B. Nushi, E. Kamar, W. S. Lasecki, D. S. Weld, and E. Horvitz, *Beyond accuracy: The role of mental models in human-ai team performance*, *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing* **7** (Oct., 2019) 2–11.

[105] G. Bansal, B. Nushi, E. Kamar, D. S. Weld, W. S. Lasecki, and E. Horvitz, *Updates in human-ai teams: Understanding and addressing the performance/compatibility tradeoff*, *Proceedings of the AAAI Conference on Artificial Intelligence* **33** (Jul., 2019) 2429–2437.

[106] Y. Zhang, Q. V. Liao, and R. K. E. Bellamy, *Effect of confidence and explanation on accuracy and trust calibration in ai-assisted decision making*, in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, (New York, NY, USA), p. 295–305, Association for Computing Machinery, 2020.

[107] S. Amershi, D. Weld, M. Vorvoreanu, A. Fourney, B. Nushi, P. Collisson, J. Suh, S. Iqbal, P. N. Bennett, K. Inkpen, J. Teevan, R. Kikin-Gil, and E. Horvitz, *Guidelines for human-ai interaction*, in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, (New York, NY, USA), p. 1–13, Association for Computing Machinery, 2019.

[108] B. Wilder, E. Horvitz, and E. Kamar, *Learning to complement humans*, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence* (Jul, 2020).

[109] T. Chakraborti and S. Kambhampati, *Algorithms for the greater good! on mental modeling and acceptable symbiosis in human-ai collaboration*, 2018.

[110] M. Kay, S. N. Patel, and J. A. Kientz, *How good is 85%? a survey tool to connect classifier evaluation to acceptability of accuracy*, in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, (New York, NY, USA), p. 347–356, Association for Computing Machinery, 2015.

[111] M. Yin, J. Wortman Vaughan, and H. Wallach, *Understanding the effect of accuracy on trust in machine learning models*, in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, (New York, NY, USA), p. 1–12, Association for Computing Machinery, 2019.

[112] R. Zhang, N. J. McNeese, G. Freeman, and G. Musick, *"an ideal human": Expectations of ai teammates in human-ai teaming*, *Proc. ACM Hum.-Comput. Interact.* **4** (jan, 2021).

[113] C. Rastogi, Y. Zhang, D. Wei, K. R. Varshney, A. Dhurandhar, and R. Tomsett, *Deciding fast and slow: The role of cognitive biases in ai-assisted decision-making*, *Proc. ACM Hum.-Comput. Interact.* **6** (apr, 2022).

[114] M. Bilgic and R. J. Mooney, *Explaining recommendations: Satisfaction vs. promotion*, in *Beyond personalization workshop, IUI*, vol. 5, p. 153, 2005.

[115] M. T. Ribeiro, S. Singh, and C. Guestrin, *"why should i trust you?": Explaining the predictions of any classifier*, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, (New York, NY, USA), p. 1135–1144, Association for Computing Machinery, 2016.

[116] S. M. Lundberg and S.-I. Lee, *A unified approach to interpreting model predictions*, in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[117] D. Wang, Q. Yang, A. Abdul, and B. Y. Lim, *Designing theory-driven user-centric explainable ai*, in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, (New York, NY, USA), p. 1–15, Association for Computing Machinery, 2019.

[118] H. Kaur, H. Nori, S. Jenkins, R. Caruana, H. Wallach, and J. Wortman Vaughan, *Interpreting interpretability: Understanding data scientists' use of interpretability tools for machine learning*, in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, (New York, NY, USA), p. 1–14, Association for Computing Machinery, 2020.

[119] B. M. Muir, *Trust between humans and machines, and the design of decision aids*, *International Journal of Man-Machine Studies* **27** (1987), no. 5 527–539.

[120] K. Yu, S. Berkovsky, R. Taib, D. Conway, J. Zhou, and F. Chen, *User trust dynamics: An investigation driven by differences in system performance*, in *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, IUI '17, (New York, NY, USA), p. 307–317, Association for Computing Machinery, 2017.

[121] J. Kunkel, T. Donkers, L. Michael, C.-M. Barbu, and J. Ziegler, *Let me explain: Impact of personal and impersonal explanations on trust in recommender systems*, in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, (New York, NY, USA), p. 1–12, Association for Computing Machinery, 2019.

[122] Z. Lu and M. Yin, *Human reliance on machine learning models when performance feedback is limited: Heuristics and risks*, in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, (New York, NY, USA), Association for Computing Machinery, 2021.

[123] R. Kocielnik, S. Amershi, and P. N. Bennett, *Will you accept an imperfect ai? exploring designs for adjusting end-user expectations of ai systems*, in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, (New York, NY, USA), p. 1–14, Association for Computing Machinery, 2019.

[124] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, *Retinaface: Single-stage dense face localisation in the wild*, *CoRR* **abs/1905.00641** (2019) [arXiv:1905.0064].

[125] S. Yang, P. Luo, C. C. Loy, and X. Tang, *Wider face: A face detection benchmark*, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[126] P. Sun, J. Cao, Y. Jiang, R. Zhang, E. Xie, Z. Yuan, C. Wang, and P. Luo, *Transtrack: Multiple object tracking with transformer*, 2020.

[127] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, *End-to-end object detection with transformers*, in *Computer Vision – ECCV 2020* (A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds.), (Cham), pp. 213–229, Springer International Publishing, 2020.

[128] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, *Deformable detr: Deformable transformers for end-to-end object detection*, *arXiv preprint arXiv:2010.04159* (2020).

[129] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, *Trackformer: Multi-object tracking with transformers*, in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June, 2022.

[130] P. Chu, J. Wang, Q. You, H. Ling, and Z. Liu, *Transmot: Spatial-temporal graph transformer for multiple object tracking*, 2021.

[131] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, *Attention is all you need*, in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[132] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, *MOT16: A benchmark for multi-object tracking*, *arXiv:1603.00831 [cs]* (Mar., 2016). arXiv: 1603.00831.

[133] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, *Mots: Multi-object tracking and segmentation*, *arXiv:1902.03604[cs]* (2019). arXiv: 1902.03604.

[134] A. Dave, T. Khurana, P. Tokmakov, C. Schmid, and D. Ramanan, *Tao: A large-scale benchmark for tracking any object*, in *European Conference on Computer Vision*, 2020.

[135] A. Dutta and A. Zisserman, *The via annotation software for images, audio and video*, in *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, (New York, NY, USA), p. 2276–2279, Association for Computing Machinery, 2019.

[136] C. Vondrick, D. J. Patterson, and D. Ramanan, *Efficiently scaling up crowdsourced video annotation - A set of best practices for high quality, economical video labeling*, *Int. J. Comput. Vis.* **101** (2013), no. 1 184–204.

[137] J. Yuen, B. Russell, C. Liu, and A. Torralba, *Labelme video: Building a video database with human annotations*, in *2009 IEEE 12th International Conference on Computer Vision*, pp. 1451–1458, 2009.

[138] I. Kavasidis, S. Palazzo, R. D. Salvo, D. Giordano, and C. Spampinato, *An innovative web-based collaborative platform for video annotation*, *Multim. Tools Appl.* **70** (2014), no. 1 413–432.

[139] J. Juran, F. Taylor, W. Shewhart, E. Deming, P. Crosby, K. Ishikawa, A. Feigenbaum, G. Taguchi, and E. Goldratt, *Quality control*, *Joseph M. Juran: Critical Evaluations in Business and Management* **1** (2005) 50.

[140] A. Neubeck and L. Van Gool, *Efficient non-maximum suppression*, in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 3, pp. 850–855, 2006.

[141] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata, *Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median*, *Journal of Experimental Social Psychology* **49** (2013), no. 4 764–766.

[142] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, *ImageNet: A large-scale hierarchical image database*, in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, (Miami, FL), pp. 248–255, IEEE, June, 2009.

[143] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, *Learning transferable visual models from natural language supervision*, 2021.

[144] S. Feiner, B. MacIntyre, T. Höllerer, and A. Webster, *A touring machine: Prototyping 3d mobile augmented reality systems for exploring the urban environment*, *Personal Technologies* **1** (1997) 208–217.

[145] T. T. M. Tran, S. Brown, O. Weidlich, M. Billinghurst, and C. Parker, *Wearable augmented reality: Research trends and future directions from three major venues*, *IEEE Transactions on Visualization and Computer Graphics* (2023).

[146] R. Gal, L. Shapira, E. Ofek, and P. Kohli, *Flare: Fast layout for augmented reality applications*, in *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 207–212, 2014.

[147] B. Nuernberger, E. Ofek, H. Benko, and A. D. Wilson, *Snaptoreality: Aligning augmented reality to the real world*, in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, (New York, NY, USA), p. 1233–1244, Association for Computing Machinery, 2016.

[148] "Hololens spatial mapping." `https://learn.microsoft.com/en-us/windows/mixed-reality/design/spatial-mapping`, 2017. [Accessed: 2024-01-30].

[149] A. Hettiarachchi and D. Wigdor, *Annexing reality: Enabling opportunistic use of everyday objects as tangible proxies in augmented reality*, in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, (New York, NY, USA), p. 1957–1967, Association for Computing Machinery, 2016.

[150] Y.-T. Yue, Y.-L. Yang, G. Ren, and W. Wang, *Scenectrl: Mixed reality enhancement via efficient scene editing*, in *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, UIST '17, (New York, NY, USA), p. 427–436, Association for Computing Machinery, 2017.

[151] D. Lindlbauer and A. D. Wilson, *Remixed reality: Manipulating space and time in augmented reality*, in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, (New York, NY, USA), p. 1–13, Association for Computing Machinery, 2018.

[152] T. Tahara, T. Seno, G. Narita, and T. Ishikawa, *Retargetable ar: Context-aware augmented reality in indoor scenes based on 3d scene graph*, in *2020 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 249–255, 2020.

[153] R. Suzuki, R. H. Kazi, L.-Y. Wei, S. DiVerdi, W. Li, and D. Leithinger, *Realitysketch: Embedding responsive graphics and visualizations in ar with dynamic sketching*, in *Adjunct Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, UIST '20 Adjunct, (New York, NY, USA), p. 135–138, Association for Computing Machinery, 2020.

[154] M. Kari, R. Schütte, and R. Sodhi, *Scene responsiveness for visuotactile illusions in mixed reality*, in *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, (New York, NY, USA), Association for Computing Machinery, 2023.

[155] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik, *Lerf: Language embedded radiance fields*, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19729–19739, 2023.

[156] A. Sharma, L. Yoffe, and T. Höllerer, *OCTO+: A Suite for Automatic Open-Vocabulary Object Placement in Mixed Reality*, in *2024 IEEE International Conference on Artificial Intelligence and eXtended and Virtual Reality (AIxVR)*, (Los Angeles, CA, USA), pp. 157–165, IEEE, Jan., 2024.

[157] B. Curless and M. Levoy, *A Volumetric Method for Building Complex Models from Range Images*. Association for Computing Machinery, New York, NY, USA, 1 ed., 2023.

[158] M. Cherti, R. Beaumont, R. Wightman, M. Wortsman, G. Ilharco, C. Gordon, C. Schuhmann, L. Schmidt, and J. Jitsev, *Reproducible scaling laws for contrastive language-image learning*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2818–2829, 2023.

[159] "Git." `https://git-scm.com/`. [Accessed: 2024-02-04].

[160] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, *Slam++: Simultaneous localisation and mapping at the level of objects*, in *2013 IEEE Conference on Computer Vision and Pattern Recognition*.

[161] L. Chen, W. Tang, N. W. John, T. R. Wan, and J. J. Zhang, *Context-Aware Mixed Reality: A Learning-Based Framework for Semantic-Level Interaction*, *Computer Graphics Forum* **39** (2020), no. 1 484–496.

[162] G. Narita, T. Seno, T. Ishikawa, and Y. Kaji, *PanopticFusion: Online Volumetric Semantic Mapping at the Level of Stuff and Things*, Sept., 2019. arXiv:1903.01177 [cs].

[163] L. Schmid, J. Delmerico, J. Schönberger, J. Nieto, M. Pollefeys, R. Siegwart, and C. Cadena, *Panoptic Multi-TSDFs: a Flexible Representation for Online Multi-resolution Volumetric Mapping and Long-term Dynamic Scene Consistency*, in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 8018–8024, May, 2022. arXiv:2109.10165 [cs].

[164] K. Č. Pucihar, V. Geroimenko, and M. Kljun, *Fuse: Towards ai-based future services for generating augmented reality experiences*, in *Augmented Reality and Artificial Intelligence: The Fusion of Advanced Technologies*, pp. 285–306. Springer, 2023.

[165] L. Yoffe, A. Sharma, and T. Höllerer, *OCTOPUS: Open-vocabulary Content Tracking and Object Placement Using Semantic Understanding in Mixed Reality*, in *2023 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, (Sydney, Australia), pp. 587–588, IEEE, Oct., 2023.

[166] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, *Segment anything*, *arXiv:2304.02643* (2023).

[167] F. Odom, "clip-text-decoder." `https://github.com/fkodom/clip-text-decoder`, 2022. [Accessed: 2024-02-04].

[168] W. Kim, B. Son, and I. Kim, *ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision*, in *Proceedings of the 38th International Conference on Machine Learning*, pp. 5583–5594, PMLR, July, 2021. ISSN: 2640-3498.

[169] T. Lüddecke and A. Ecker, *Image segmentation using text and image prompts*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7086–7096, June, 2022.

[170] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang, *Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection*, Mar., 2023. arXiv:2303.05499 [cs].

[171] OpenAI, J. Achiam, S. Adler, S. Agarwal, and ..., *GPT-4 Technical Report*, Mar., 2024. arXiv:2303.08774 [cs].

[172] H. Liu, C. Li, Y. Li, and Y. J. Lee, *Improved baselines with visual instruction tuning*, 2023.

[173] T. Höllerer, S. Feiner, T. Terauchi, G. Rashid, and D. Hallaway, *Exploring mars: developing indoor and outdoor user interfaces to a mobile augmented reality system*, *Computers & Graphics* **23** (1999), no. 6 779–785.

[174] S. Mann, *Wearable computing: a first step toward personal imaging*, *Computer* **30** (1997), no. 2 25–32.

[175] J. C. Spohrer, *Information in places*, *IBM Systems Journal* **38** (1999), no. 4 602–628.

[176] G. A. Lee, G. J. Kim, and M. Billinghurst, *Immersive authoring: What you experience is what you get (wyxiwyg)*, *Commun. ACM* **48** (jul, 2005) 76–81.

[177] J. Valentin, V. Vineet, M.-M. Cheng, D. Kim, J. Shotton, P. Kohli, M. Nießner, A. Criminisi, S. Izadi, and P. Torr, *Semanticpaint: Interactive 3d labeling and learning at your fingertips*, *ACM Trans. Graph.* **34** (nov, 2015).

[178] O. Miksik, V. Vineet, M. Lidegaard, R. Prasaath, M. Nießner, S. Golodetz, S. L. Hicks, P. Pérez, S. Izadi, and P. H. Torr, *The semantic paintbrush: Interactive 3d mapping and recognition in large outdoor spaces*, in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, (New York, NY, USA), p. 3317–3326, Association for Computing Machinery, 2015.

[179] B. Huynh, J. Orlosky, and T. Höllerer, *In-Situ Labeling for Augmented Reality Language Learning*, in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 1606–1611, Mar., 2019. ISSN: 2642-5254.

[180] D. Agrawal, J. Lobsiger, Y. F. Bo, V. Kaufmann, and I. Armeni, *Hololabel: Augmented reality user-in-the-loop online annotation tool for as-is building information*, in *Proceedings of the 2022 European Conference on Computing in Construction*, (Torino), pp. 580 – 589, Università degli Studi di Torino, 2022.

[181] M. Kari, T. Grosse-Puppendahl, L. F. Coelho, A. R. Fender, D. Bethge, R. Schütte, and C. Holz, *Transformr: Pose-aware object substitution for composing alternate mixed realities*, in *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 69–79, 2021.

[182] X. Qian, F. He, X. Hu, T. Wang, A. Ipsita, and K. Ramani, *Scalar: Authoring semantically adaptive augmented reality experiences in virtual reality*, in *Proceedings of the 2022 CHI Conference on Human Factors in Computing*

Systems, CHI '22, (New York, NY, USA), Association for Computing Machinery, 2022.

[183] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, *et. al.*, *Flamingo: a visual language model for few-shot learning*, *Advances in neural information processing systems* **35** (2022) 23716–23736.

[184] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig, *Scaling up visual and vision-language representation learning with noisy text supervision*, in *International conference on machine learning*, pp. 4904–4916, PMLR, 2021.

[185] X. Gu, T.-Y. Lin, W. Kuo, and Y. Cui, *Open-vocabulary object detection via vision and language knowledge distillation*, *arXiv preprint arXiv:2104.13921* (2021).

[186] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl, *Language-driven semantic segmentation*, *arXiv preprint arXiv:2201.03546* (2022).

[187] Y. Rao, W. Zhao, G. Chen, Y. Tang, Z. Zhu, G. Huang, J. Zhou, and J. Lu, *Denseclip: Language-guided dense prediction with context-aware prompting*, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18082–18091, 2022.

[188] S. Peng, K. Genova, C. Jiang, A. Tagliasacchi, M. Pollefeys, T. Funkhouser, *et. al.*, *Openscene: 3d scene understanding with open vocabularies*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 815–824, 2023.

[189] J. Zhang, R. Dong, and K. Ma, *Clip-fo3d: Learning free open-world 3d scene representations from 2d dense clip*, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2048–2059, 2023.

[190] R. Ding, J. Yang, C. Xue, W. Zhang, S. Bai, and X. Qi, *Pla: Language-driven open-vocabulary 3d scene understanding*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7010–7019, 2023.

[191] J. Yang, R. Ding, Z. Wang, and X. Qi, *Regionplc: Regional point-language contrastive learning for open-world 3d scene understanding*, *arXiv preprint arXiv:2304.00962* (2023).

[192] J. Rekimoto, *Time-machine computing: a time-centric approach for the information environment*, in *Proceedings of the 12th Annual ACM Symposium on User Interface Software and Technology*, UIST '99, (New York, NY, USA), p. 45–54, Association for Computing Machinery, 1999.

[193] J. D. Denning and F. Pellacini, *MeshGit: diffing and merging meshes for polygonal modeling*, *ACM Transactions on Graphics* **32** (July, 2013) 35:1–35:10.

[194] E. Carra and F. Pellacini, *Scenegit: a practical system for diffing and merging 3d environments*, *ACM Trans. Graph.* **38** (nov, 2019).

[195] K. Lilija, H. Pohl, and K. Hornbæk, *Who put that there? temporal navigation of spatial recordings by direct manipulation*, in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, (New York, NY, USA), p. 1–11, Association for Computing Machinery, 2020.

[196] L. Zhang, A. Agrawal, S. Oney, and A. Guo, *Vrgit: A version control system for collaborative content creation in virtual reality*, in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, (New York, NY, USA), Association for Computing Machinery, 2023.

[197] F. Perteneder, E.-M. Grossauer, Y. Xu, and M. Haller, *Catch-Up 360: Digital Benefits for Physical Artifacts*, in *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '15, (New York, NY, USA), pp. 105–108, Association for Computing Machinery, Jan., 2015.

[198] M. Letter, M. Kurzweg, and K. Wolf, *Comparing Screen-Based Version Control to Augmented Artifact Version Control for Physical Objects*, in *Human-Computer Interaction – INTERACT 2023*, Lecture Notes in Computer Science, (Cham), pp. 391–415, Springer Nature Switzerland, 2023.

[199] A. R. Fender and C. Holz, *Causality-preserving Asynchronous Reality*, in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, (New York, NY, USA), pp. 1–15, Association for Computing Machinery, Apr., 2022.

[200] Q. Yu, H. Wang, S. Qiao, M. Collins, Y. Zhu, H. Adam, A. Yuille, and L.-C. Chen, *k-means mask transformer*, in *ECCV*, 2022.

[201] W. E. Lorensen and H. E. Cline, *Marching cubes: A high resolution 3d surface construction algorithm*, in *Seminal graphics: pioneering efforts that shaped the field*, pp. 347–353. 1998.

[202] A. R. Smith, *Tint fill*, in *Proceedings of the 6th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '79, (New York, NY, USA), p. 276–283, Association for Computing Machinery, 1979.

[203] Z. Feng, H. Zhan, Z. Chen, Q. Yan, X. Xu, C. Cai, B. Li, Q. Zhu, and Y. Xu, *Naruto: Neural active reconstruction from uncertain target observations*, *arXiv preprint arXiv:2402.18771* (2024).

[204] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, *Dynamic graph cnn for learning on point clouds*, *ACM Trans. Graph.* **38** (oct, 2019).

[205] Y. Li, H. Wang, Y. Duan, and X. Li, *Clip surgery for better explainability with enhancement in open-vocabulary tasks*, 2023.

[206] M. Fiala, *Artag, a fiducial marker system using digital techniques*, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, pp. 590–596 vol. 2, 2005.

[207] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, *3dmatch: Learning local geometric descriptors from rgb-d reconstructions*, in *CVPR*, 2017.

[208] "Llmperf leaderboard." `https://github.com/ray-project/llmperf-leaderboard?tab=readme-ov-file`, 2023. [Accessed: 2024-02-04].

[209] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollar, *Panoptic segmentation*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June, 2019.

[210] H. Touvron, L. Martin, K. Stone, ..., and T. Scialom, *Llama 2: Open Foundation and Fine-Tuned Chat Models*, July, 2023. arXiv:2307.09288 [cs].

[211] "Meta llama 3." `https://ai.meta.com/blog/meta-llama-3/`. [Accessed: 2024-05-07].

[212] "What is a long context window?." `https://blog.google/technology/ai/long-context-window-ai-models/`. [Accessed: 2024-03-01].

[213] J. Yang, K. Zhou, Y. Li, and Z. Liu, *Generalized out-of-distribution detection: A survey*, *arXiv preprint arXiv:2110.11334* (2021).

[214] D. J. Simons and D. T. Levin, *Change blindness*, *Trends in cognitive sciences* **1** (1997), no. 7 261–267.

[215] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, *Zero-shot text-to-image generation*, 2021.

[216] D. A. Norman and S. W. Draper, *User Centered System Design; New Perspectives on Human-Computer Interaction*. L. Erlbaum Associates Inc., USA, 1986.

[217] R. Fiebrink, P. R. Cook, and D. Trueman, *Human model evaluation in interactive supervised learning*, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, (New York, NY, USA), p. 147–156, Association for Computing Machinery, 2011.

[218] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Imagenet classification with deep convolutional neural networks*, in *Advances in Neural Information Processing Systems* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), vol. 25, Curran Associates, Inc., 2012.

[219] M. D. Zeiler and R. Fergus, *Visualizing and understanding convolutional networks*, in *Computer Vision – ECCV 2014* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), (Cham), pp. 818–833, Springer International Publishing, 2014.

[220] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, *Going deeper with convolutions*, in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

[221] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, in *International Conference on Learning Representations*, 2015.

[222] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, *arXiv preprint arXiv:1512.03385* (2015).

[223] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, *ImageNet Large Scale Visual Recognition Challenge*, *International Journal of Computer Vision (IJCV)* **115** (2015), no. 3 211–252.

[224] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, *ImageNet: A Large-Scale Hierarchical Image Database*, in *CVPR09*, 2009.

[225] K. He, X. Zhang, S. Ren, and J. Sun, *Delving deep into rectifiers: Surpassing human-level performance on imagenet classification*, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December, 2015.

[226] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, *Emerging properties in self-supervised vision transformers*, in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.