A Foundation Model for Cell Segmentation

Uriah Israel[1,3†], Markus Marks[2,3†], Rohit Dilip[3†], Qilin Li[2], Morgan Schwartz[1], Elora Pradhan[1], Edward Pao[1], Shenyi Li[1], Alexander Pearson-Goulart[1], Pietro Perona[2,3], Georgia Gkioxari[3], Ross Barnowski[1], Yisong Yue[3], David Van Valen[1,4*]

[1*]Division of Biology and Biological Engineering, Caltech.
[2]Division of Engineering and Applied Science, Caltech.
[3]Division of Computing and Mathematical Science, Caltech.
[4]Howard Hughes Medical Institute.

*Corresponding author(s). E-mail(s): vanvalen@caltech.edu;
Contributing authors: ulisrael@caltech.edu; marks@caltech.edu; rdilip@caltech.edu; qli2@caltech.edu; msschwartz@caltech.edu; epradhan@caltech.edu; epao@caltech.edu; sli5@caltech; pearsongoulart@gmail.com; perona@caltech.edu; georgia@caltech.edu; rossbar@caltech.edu; yyue@caltech.edu;
†These authors contributed equally to this work.

Abstract

Cells are the fundamental unit of biological organization, and identifying them in imaging data - cell segmentation - is a critical task for various cellular imaging experiments. While deep learning methods have led to substantial progress on this problem, models that have seen wide use are specialist models that work well for specific domains. Methods that have learned the general notion of "what is a cell" and can identify them across different domains of cellular imaging data have proven elusive. In this work, we present CellSAM, a foundation model for cell segmentation that generalizes across diverse cellular imaging data. CellSAM builds on top of the Segment Anything Model (SAM) by developing a prompt engineering approach to mask generation. We train an object detector, CellFinder, to automatically detect cells and prompt SAM to generate segmentations. We show that this approach allows a single model to achieve state-of-the-art performance for segmenting images of mammalian cells (in tissues and cell culture), yeast, and bacteria collected with various imaging modalities. To enable accessibility, we integrate CellSAM into DeepCell Label to further accelerate human-in-the-loop labeling strategies for cellular imaging data. A deployed version of CellSAM is available at https://label-dev.deepcell.org/.

Keywords: cell segmentation, object detection, deep learning, foundation model

1 Introduction

Accurate cell segmentation is crucial for quantitative analysis and interpretation of various cellular imaging experiments. Modern spatial genomics assays can produce data on the location and abundance of 10^1-10^2 protein species and 10^2-10^4 RNA species simultaneously in living and fixed tissues[1–5]. These data shed light on the biology of healthy and diseased tissues but are challenging to interpret. Cell segmentation enables these data to be converted to interpretable tissue maps of protein localization and transcript abundances. Similarly, live-cell imaging provides insight into dynamic phenomena in bacterial and mammalian cell biology. Mechanistic insights into critical phenomena such as the mechanical behavior of the bacterial cell wall[6,7], information transmission in cell signaling pathways[8–11], heterogeneity in immune cell behavior

during immunotherapy[12], and the morphodynamics of development[13] have been gained by analyzing live-cell imaging data. Like their tissue counterparts, cell segmentation is also a key challenge for these experiments, as cells must be segmented and tracked to create temporally consistent records of cell behavior that can be queried at scale.

Significant progress has been made in recent years on the problem of cell segmentation, primarily driven by advances in deep learning[14]. Progress in this space has occurred mainly in two distinct but related directions. In the first direction is work that explores the space of deep learning methods that generalize well to cellular imaging data. This includes explorations on deep learning architectures that generalize as well as the representations used to present the notion of what a cell is to a given model[15–21]. The second direction is to work on improving labeling methodology. Cell segmentation is a variant of the instance segmentation problem, which requires pixel-level labels for every object in an image. Creating these labels can be expensive ($10^{-2} - 10^{0}$ USD/label)[18,22], which provides an incentive to reduce the marginal cost of labeling. A recent improvement to labeling methodology has been human-in-the-loop labeling, where labelers correct model errors rather than produce labels from scratch[16,18,23].

Despite this progress, two critical gaps still need to be addressed. The first is a cell segmentation method that can generalize across diverse cellular images. Existing methods are primarily specialist models - design choices in cellular representation restrict their accuracy to a specific domain. For example, Mesmer's[18] representation for a cell (cell centroid and boundary) enables good performance in tissue images but would be a poor choice for elongated bacterial cells. Similar trade-offs in representations exist for the current collection of Cellpose models, necessitating the creation of a model zoo[16]. The second gap is new labeling methodologies that can further reduce the marginal cost of cell labeling. While this cost has been reduced substantially by recent work[18,23], reducing this further could increase the amount of labeled imaging data by orders of magnitude.

Recent work in machine learning on foundation models holds promise for providing a complete solution. Foundation models are large deep neural network models (typically transformers[24]) trained on a large amount of data in a self-supervised fashion with supervised fine-tuning on one or several tasks[25]. Foundation models include the GPT[26,27] family of models, which have proven transformative for natural language processing[25] and have been used in other domains, such as biological sequences[28]. These successes have inspired similar efforts in computer vision. The Vision Transformer[29] was introduced in 2020 and has since been used as the basis architecture for a collection of vision foundation models[30–34]. One recent foundation model well suited to cellular image analysis needs is the Segment Anything Model (SAM)[35]. This model uses a Vision Transformer (ViT) to extract information-rich features from raw images. These features are then directed to a module that generates instance masks based on prompts, which can be either spatial (e.g., an object centroid or bounding box) or semantic (e.g., an object's visual description). Notably, the promptable nature of SAM enabled scalable dataset construction, as preliminary versions of SAM allowed labelers to generate accurate instance masks with 1-2 clicks. The final version of SAM was trained on a dataset of 1 billion masks over 11 million images and demonstrated strong performance on various zero-shot learning tasks. Recent work has attempted to apply SAM to problems in biological and medical imaging, including medical image segmentation[36–38], lesion detection in dermatological images[39,40], nuclear segmentation in H&E images[41,42] and fine-tuned SAM on cellular image data for use in the Napari software package[43].

While promising, these studies reported challenges adapting SAM to these new use cases[36,43]. These challenges include reduced performance and uncertain boundaries when transitioning from natural to medical images. Cellular images contain additional complications – they can involve different imaging modalities (e.g., phase microscopy vs. fluorescence microscopy), thousands of objects in a field of view (as opposed to dozens in a natural image), uncertain and noisy boundaries (artifacts of projecting 3D objects into a 2D plane)[43]. In addition to these challenges, SAM's default prompting strategy does not allow for accurate inference for cellular images. Currently, the automated prompting of SAM uses a uniform grid of points to generate masks, an approach poorly suited to cellular images given the wide variation of cell densities. More precise prompting (e.g., a bounding box or mask) requires prior knowledge of cell locations. This creates a weak tautology - SAM can find the cells provided it knows a priori where they are. This limitation makes it challenging for SAM to serve as a foundation model for cell segmentation - it can accelerate labeling but still requires human input for inference. A solution to this problem would enable SAM-like models to serve as foundation models and knowledge engines, as they could accelerate the generation of labeled data, learn from them, and make that knowledge accessible to life scientists via inference.

In this work, we developed CellSAM, a foundation model for cell segmentation (Fig. 1). CellSAM extends the SAM methodology to perform automated cellular instance segmentation. To achieve this, we first assembled a comprehensive dataset for cell segmentation spanning five different morphological archetypes. To automate inference with SAM, we took a prompt engineering approach and explored the best ways to prompt SAM to generate high-quality masks. We observed that bounding boxes consistently generated high-quality masks compared to alternative approaches. We further identified a compute-efficient method to fine-tune SAM to achieve even better performance. To facilitate automated inference through prompting, we developed CellFinder, a transformer-based object detector that uses the Anchor DETR framework. Within CellSAM, CellFinder and SAM shares the same ViT backbone; the bounding boxes generated by CellFinder are then used as prompts for SAM, enumerating masks for all the cells in an image. We trained CellSAM on a large, diverse corpus of cellular imaging data, enabling it to achieve state-of-the-art (SOTA) performance on nine datasets. We also evaluated CellSAM's zero-shot performance using a held-out dataset[44], demonstrating that it outperforms existing methods for zero-shot segmentation. The datasets described in this work are available at https://deepcell.readthedocs.io/en/master/data-gallery/; a deployed version of CellSAM is available at our lab's web portal https://deepcell.org.

2 Results

2.1 Construction of a dataset for general cell segmentation

A significant challenge with existing cellular segmentation methods is their inability to generalize across various imaging modalities and cell morphologies. To address this, we curated a dataset from the literature containing 2D images of various cell morphologies (mammalian cells in tissues and adherent cell culture, yeast cells, bacterial cells, and mammalian cell nuclei) and imaging modalities (fluorescence, brightfield, phase contrast, hematoxylin & eosin staining, and mass cytometry imaging). For each ingested dataset, we inspected them for data leaks between training and testing splits and removed them when present. Our final dataset consisted of TissueNet[18], DeepBacs[45], BriFiSeg[46], Cellpose[15,16],

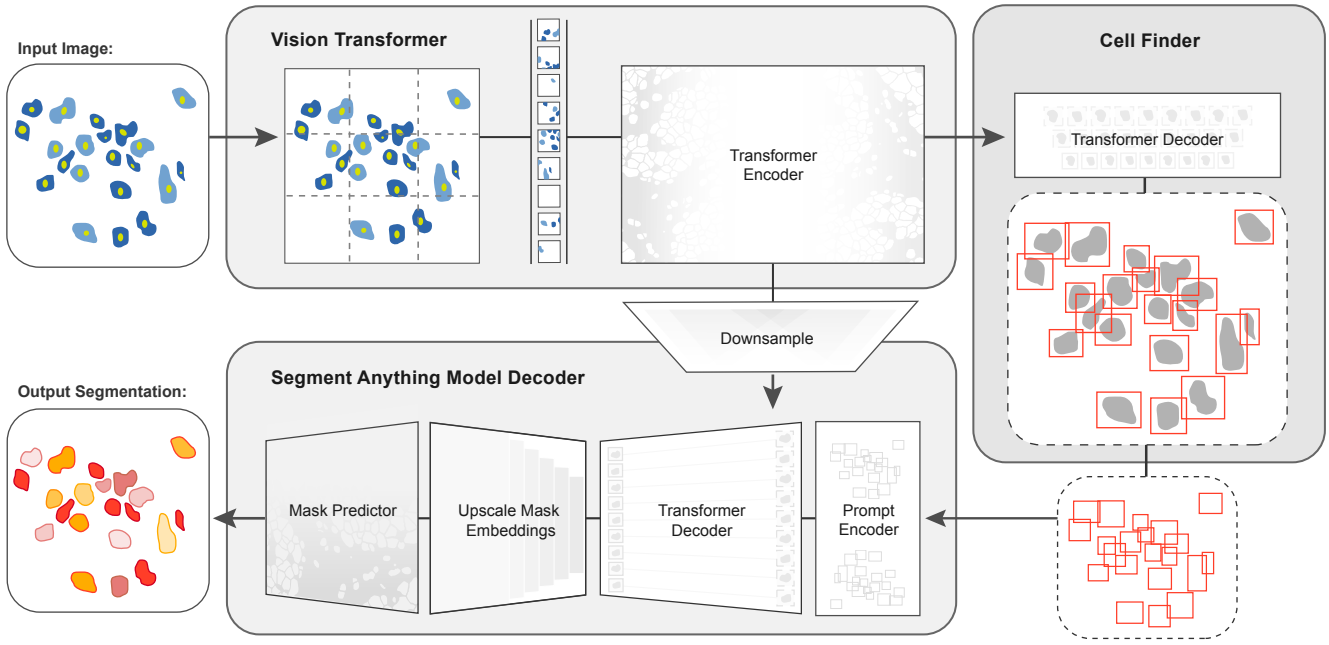

Fig. 1 CellSAM: a foundational model for cell segmentation. CellSAMcombines SAM's mask generation and labeling capabilities with an object detection model to achieve automated inference. Input images are divided into regularly sampled patches and passed through a transformer encoder (e.g., a ViT) to generate information-rich image features. These image features are then sent to two downstream modules. The first module, CellFinder, decodes these features into bounding boxes using a transformer-based encoder-decoder pair. The second module combines these image features with prompts to generate masks using SAM's mask decoder. CellSAM integrates these two modules using the bounding boxes generated by CellFinder as prompts for SAM. CellSAM is trained in two stages, using the pre-trained SAM model weights as a starting point. In the first stage, we train the ViT and the CellFinder model together on the object detection task. This yields an accurate CellFinder but results in a distribution shift between the ViT and SAM's mask decoder. The second stage closes this gap by fixing the ViT and SAM mask decoder weights and fine-tuning the remainder of the SAM model (i.e., the model neck) using ground truth bounding boxes and segmentation labels.

Omnipose[47,48], YeastNet[49], YeaZ[50], the 2018 Kaggle Data Science Bowl dataset (DSB)[51], and an internally collected dataset of phase microscopy images across eight mammalian cell lines (Phase400). For evaluation, we group these datasets into four types: Tissue, Cell Culture, Bacteria, and Yeast. As the DSB[51] comprises cell nuclei that span several of these types, we evaluate it separately and refer to it as Nuclear. While our method focuses on whole-cell segmentation, we included DSB[51] because cell nuclei are often used as a surrogate when the information necessary for whole-cell segmentation (e.g., cell membrane markers) is absent from an image. A summary of the dataset is shown in Figure 2a. To evaluate CellSAM's zero-shot performance, we used a held-out LIVECell[44] dataset. A detailed description of data sources and pre-processing steps can be found in the Appendix A.

2.2 Bounding boxes are accurate prompts for cell segmentation with SAM

For accurate inference, SAM needs to be provided with approximate information about the location of cells in the form of prompts. To better engineer prompts, we first assessed SAM's ability to generate masks when provided prompts derived from ground truth labels - either point prompts (derived from the cell's center of mass) or bounding box prompts. For these tests, we used the pre-trained model weights that were publicly released[35]. Our benchmarking results are shown in Figure 2b and revealed that bounding boxes had significantly higher zero-shot performance than point prompting, although both approaches struggled with Tissue imaging data. To improve SAM's mask generation ability for cellular image data, we explored fine-tuning SAM on our compiled data to help it bridge the gap from natural to cellular images. During these fine-tuning experiments, we observed that fine-tuning all of SAM was unnecessary; instead, we only needed

to fine-tune the layers connecting SAM's ViT to its decoder, the model neck, to achieve good performance. All other layers can be frozen. Fine-tuning SAM in this fashion led to a model capable of generating high-quality cell masks when prompted by ground truth bounding boxes, as seen in Figure 2b.

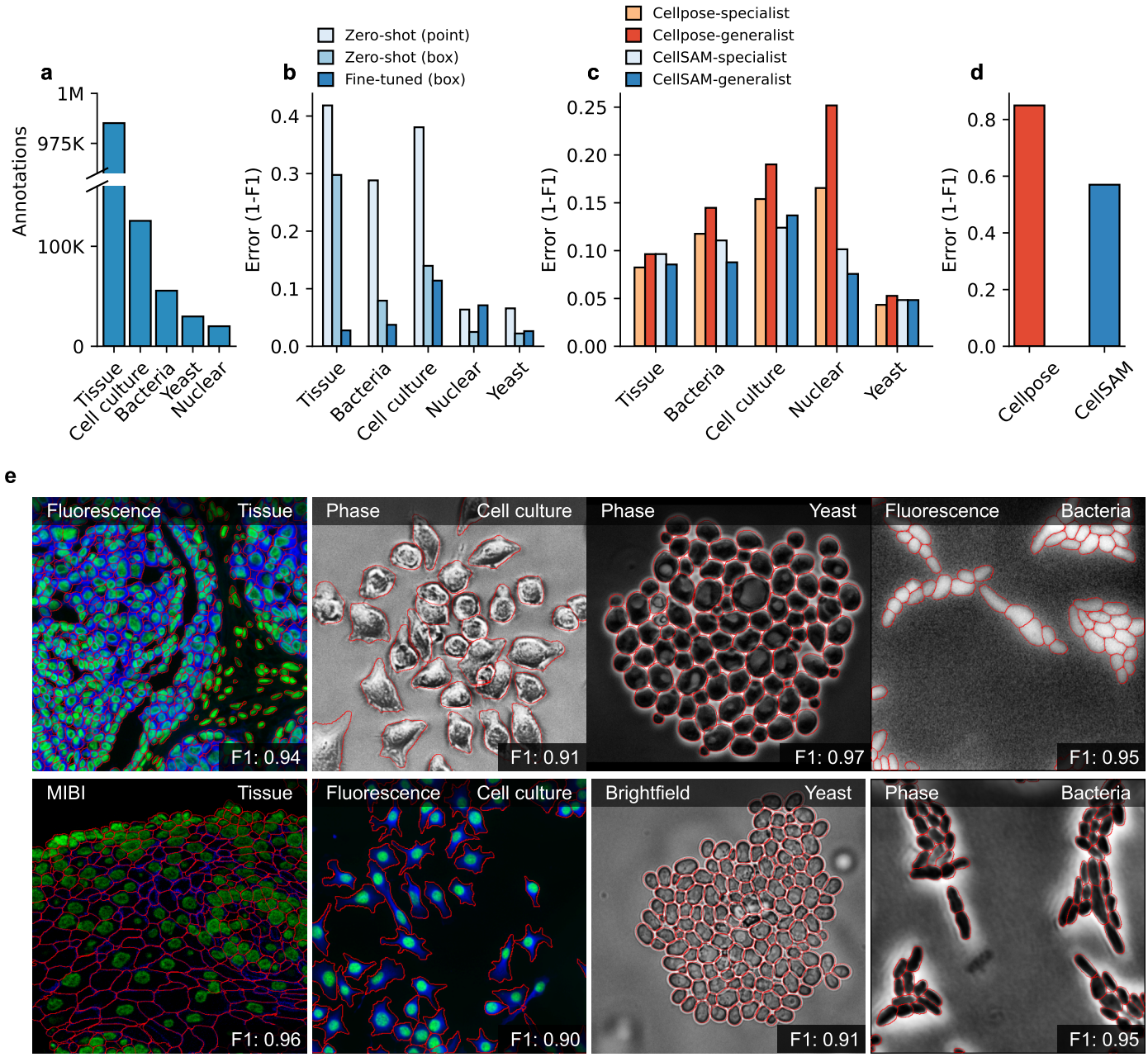

Fig. 2 CellSAM is a strong generalist model for cell segmentation. a) For training and evaluating CellSAM, we curated a diverse cell segmentation dataset from the literature. The number of annotated cells is given for each data type. Nuclear refers to a heterogeneous dataset (DSB)[51] containing nuclear segmentation labels. b) Zero-shot (ZS) and fine-tuned mask generation error (1- F1 score) for SAM when using point and bounding box prompts. All prompting in this figure was done with ground truth prompts. The best performance is achieved with bounding box prompts and fine-tuning. c) Segmentation performance for CellSAM and Cellpose on different data types. We compare the segmentation error (1-F1) for models that were trained as specialists (e.g., on one dataset) or generalists (the full dataset). Models were trained for a similar number of steps across all datasets. We observed that CellSAM-generalhas a lower error than Cellpose-general on almost all tested datasets. Further, we observed that generalist training improved CellSAM's performance over specialist training; the reverse was true for Cellpose. d) Zero-shot performance of CellSAM-general and Cellpose-General on the LIVECell dataset. Here, we show greater than 4x segmentation performance on an unseen dataset. e) Qualitative results of CellSAM segmentations for different data and imaging modalities. Predicted segmentations are outlined in red.

2.3 CellFinder and CellSAM enable accurate and automated cell segmentation

Given that bounding box prompts yield accurate segmentation masks from SAM across various datasets, we sought to develop an object detector that could generate prompts for SAM in an automated fashion. Given that our zero-shot experiments demonstrated that ViT features can form robust internal representations of cellular images, we reasoned we could build an object detector on top of the image features generated by SAM's ViT. Previous work has explored this space and demonstrated that ViT backbones can achieve SOTA performance on natural images[52,53]. For our object detection module, we use the Anchor DETR framework[54], using the same ViT backbone as the SAM module; we call this object detection module CellFinder. Anchor DETR is well suited for object detection in cellular images because it formulates object detection as a set prediction task. This allows it to - in theory - perform cell segmentation in images that are densely packed or contain overlapping objects, common occurrences in cellular imaging data. These failure modes are challenging to address with existing methods. Bounding box methods (e.g., the R-CNN family[55,56]) rely on non-maximum suppression, leading to poor performance in this regime. Methods that frame cell segmentation as a dense, pixel-wise prediction task (e.g., Mesmer[18] and Cellpose[15]) assume that each pixel can be uniquely assigned to a single cell and cannot handle overlapping objects.

We train CellSAM in two stages; the full details can be found in Appendices B. In the first stage, we train CellFinder on the object detection task. We convert the ground truth cell masks into bounding boxes and train the ViT backbone and the CellFinder module. Once CellFinder is trained, we freeze the model weights of the ViT and fine-tune the SAM module as described above. This accounts for the distribution shifts in the ViT features that occur during the CellFinder training. Once training is complete, we use CellFinder to prompt SAM's mask decoder. We refer to the collective method as CellSAM; Figure 1 outlines an image's full path through CellSAM during inference. We benchmark CellSAM's performance using a suite of metrics (Figure 2c and 2d and Supplemental Figure S2) and find that it outperforms Cellpose models trained on comparable datasets. We highlight two features of our benchmarking analyses below.

- **CellSAM is a strong generalist model**. Generalization across cell morphologies and imaging datasets has been a significant challenge for deep learning-based cell segmentation algorithms. To evaluate CellSAM's generalization capabilities, we compared its performance to CellSAM and Cellpose models trained as specialists (e.g., on a single dataset) or generalists (e.g., on the entire dataset). Consistent with the literature, we observed that Cellpose's performance degraded when trained as a generalist (Figure 2c), as specialist Cellpose models had a higher F1 score across all datasets. We observed that the reverse was true for CellSAM; the F1 score remained the same or improved in four of the five data categories and across seven of the nine datasets (Figure 2 and Supplemental Figure S2).
- **CellSAM achieves SOTA zero-shot performance**. To further evaluate CellSAM's capacity for generalization, we evaluated its performance on an entirely unseen dataset, LIVECell[44], without further fine-tuning. When compared against the Cellpose-generalist model, we find that CellSAM's zero-shot segmentation performance is considerably better, albeit still not accurate enough to be used in real-world settings. We note that some of the poor reported performance is due to label errors in the LIVECell dataset[16].

3 Discussion

Cell segmentation is a critical task for cellular imaging experiments. While deep learning methods have made substantial progress in recent years, there remains a need for methods that can generalize across diverse images and further reduce the marginal cost of image labeling. In this work, we sought to meet these needs by developing CellSAM, a foundation model for cell segmentation. Transformer-based methods for cell segmentation are showing promising performance. CellSAM builds on these works by integrating the mask generation capabilities of SAM with transformer-based object detection to empower both scalable image labeling and automated inference. We trained CellSAM on a diverse dataset curated from the literature. Our benchmarking demonstrated that CellSAM achieves SOTA performance on cell segmentation and that this performance is aided by our attempts to create a general segmentation model. Given its utility in image labeling and accuracy during inference, we believe CellSAM is a valuable contribution to the field and will help create the data infrastructure required for cellular imaging's AI-powered future.

The work described here has importance beyond aiding life scientists with cell segmentation. First, foundation models are immensely useful for natural language and vision tasks and hold similar promise for the life sciences - provided they are suitably adapted to this new domain. We can see several uses for CellSAM that might be within reach of future work. First, given its generalization capabilities, it is likely that CellSAM has learned a general representation for the notion of "cells" used to query imaging data. These representations might serve as an interface between imaging data and other modalities (e.g., single-cell RNA Sequencing), provided there is suitable alignment between cellular representations for each domain[57,58]. Second, much like what has occurred with natural images, we foresee that the integration of natural language labels in addition to cell-level labels might lead to vision-language models capable of generating human-like descriptors of cellular images with entity-level resolution[33]. Third, the generalization capabilities may enable the standardization of cellular image analysis pipelines across all the life sciences. If the accuracy is sufficient, microbiologists and tissue biologists could use the same collection of foundation models for interpreting their imaging data even for challenging experiments[59,60]. Last, new efforts seek to generate AI scientists capable of generating hypotheses and exploring them through the design and execution of new experiments[61]. Foundation models like CellSAM could contribute to this vision by serving as this scientist's "eyes", converting complex imaging data to structured knowledge that can be operationalized.

While the work presented here highlights the potential foundation models hold for cellular image analysis, much work remains to be done for this future to manifest. Extension of this methodology to 3D imaging data is essential; recent work on memory-efficient attention kernels[62] will aid these efforts. Exploring how to enable foundation models to leverage the full information content of images (e.g., multiple stains, temporal information for movies, etc.) is an essential avenue of future work. Expanding the space of labeled data remains a priority - this includes images of perturbed cells and cells with more challenging morphologies (e.g., neurons). Data generated by pooled optical screens[63] may synergize well with the data needs of foundation models. Compute-efficient fine-tuning strategies must be developed to enable flexible adaptation to new image domains. Lastly, prompt engineering is a critical area of future work, as it is critical to maximizing model performance. The work we presented here can be thought of as prompt engineering, as we leverage CellFinder to produce

bounding box prompts for SAM. As more challenging labeled datasets are incorporated, the nature of the "best" prompts will likely evolve. Finding the best prompts for these new data, rather than the best vision pipelines, is a task that will likely fall on both the computer vision and life science communities.

References

[1] G. Palla, D. S. Fischer, A. Regev, and F. J. Theis, "Spatial components of molecular tissue biology," *Nature Biotechnology*, vol. 40, no. 3, pp. 308–318, 2022.

[2] J. R. Moffitt, E. Lundberg, and H. Heyn, "The emerging landscape of spatial profiling technologies," *Nature Reviews Genetics*, vol. 23, no. 12, pp. 741–759, 2022.

[3] L. Moses and L. Pachter, "Museum of spatial transcriptomics," *Nature Methods*, vol. 19, no. 5, pp. 534–546, 2022.

[4] J. W. Hickey, E. K. Neumann, A. J. Radtke, J. M. Camarillo, R. T. Beuschel, A. Albanese, E. McDonough, J. Hatler, A. E. Wiblin, J. Fisher *et al.*, "Spatial mapping of protein composition and tissue organization: a primer for multiplexed antibody-based imaging," *Nature methods*, vol. 19, no. 3, pp. 284–295, 2022.

[5] J. Ko, M. Wilkovitsch, J. Oh, R. H. Kohler, E. Bolli, M. J. Pittet, C. Vinegoni, D. B. Sykes, H. Mikula, R. Weissleder *et al.*, "Spatiotemporal multiplexed immunofluorescence imaging of living cells and tissues with bioorthogonal cycling of fluorescent probes," *Nature Biotechnology*, vol. 40, no. 11, pp. 1654–1662, 2022.

[6] S. Wang, L. Furchtgott, K. C. Huang, and J. W. Shaevitz, "Helical insertion of peptidoglycan produces chiral ordering of the bacterial cell wall," *Proceedings of the National Academy of Sciences*, vol. 109, no. 10, pp. E595–E604, 2012.

[7] E. R. Rojas, G. Billings, P. D. Odermatt, G. K. Auer, L. Zhu, A. Miguel, F. Chang, D. B. Weibel, J. A. Theriot, and K. C. Huang, "The outer membrane is an essential load-bearing element in gram-negative bacteria," *Nature*, vol. 559, no. 7715, pp. 617–621, 2018.

[8] A. S. Hansen and E. K. O'Shea, "Limits on information transduction through amplitude and frequency regulation of transcription factor activity," *Elife*, vol. 4, p. e06559, 2015.

[9] A. S. Hansen and E. K. O'shea, "Promoter decoding of transcription factor dynamics involves a trade-off between noise and control of gene expression," *Molecular systems biology*, vol. 9, no. 1, p. 704, 2013.

[10] S. Tay, J. J. Hughey, T. K. Lee, T. Lipniacki, S. R. Quake, and M. W. Covert, "Single-cell nf-κb dynamics reveal digital activation and analogue information processing," *Nature*, vol. 466, no. 7303, pp. 267–271, 2010.

[11] S. Regot, J. J. Hughey, B. T. Bajar, S. Carrasco, and M. W. Covert, "High-sensitivity measurements of multiple kinase activities in live single cells," *Cell*, vol. 157, no. 7, pp. 1724–1734, 2014.

[12] M. Alieva, A. K. Wezenaar, E. J. Wehrens, and A. C. Rios, "Bridging live-cell imaging and next-generation cancer treatment," *Nature Reviews Cancer*, pp. 1–15, 2023.

[13] J. Cao, G. Guan, V. W. S. Ho, M.-K. Wong, L.-Y. Chan, C. Tang, Z. Zhao, and H. Yan, "Establishment of a morphological atlas of the caenorhabditis elegans embryo using deep-learning-based 4d segmentation," *Nature communications*, vol. 11, no. 1, p. 6254, 2020.

[14] M. Schwartz, U. Israel, X. Wang, E. Laubscher, C. Yu, R. Dilip, Q. Li, J. Mari, J. Soro, K. Yu *et al.*, "Scaling biological discovery at the interface of deep learning and cellular imaging," *Nature Methods*, vol. 20, no. 7, pp. 956–957, 2023.

[15] C. Stringer, T. Wang, M. Michaelos, and M. Pachitariu, "Cellpose: a generalist algorithm for cellular segmentation," *Nature methods*, vol. 18, no. 1, pp. 100–106, 2021.

[16] M. Pachitariu and C. Stringer, "Cellpose 2.0: how to train your own model," *Nature Methods*, pp. 1–8, 2022.

[17] U. Schmidt, M. Weigert, C. Broaddus, and G. Myers, "Cell detection with star-convex polygons," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II 11*. Springer, 2018, pp. 265–273.

[18] N. F. Greenwald, G. Miller, E. Moen, A. Kong, A. Kagel, T. Dougherty, C. C. Fullaway, B. J. McIntosh, K. X. Leow, M. S. Schwartz *et al.*, "Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning," *Nature biotechnology*, vol. 40, no. 4, pp. 555–565, 2022.

[19] R. Hollandi, A. Szkalisity, T. Toth, E. Tasnadi, C. Molnar, B. Mathe, I. Grexa, J. Molnar, A. Balind, M. Gorbe *et al.*, "nucleaizer: a parameter-free deep learning framework for nucleus segmentation using image style transfer," *Cell Systems*, vol. 10, no. 5, pp. 453–458, 2020.

[20] S. Graham, Q. D. Vu, S. E. A. Raza, A. Azam, Y. W. Tsang, J. T. Kwak, and N. Rajpoot, "Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images," *Medical image analysis*, vol. 58, p. 101563, 2019.

[21] W. Wang, D. A. Taft, Y.-J. Chen, J. Zhang, C. T. Wallace, M. Xu, S. C. Watkins, and J. Xing, "Learn to segment single cells with deep distance estimator and deep cell detector," *Computers in biology and medicine*, vol. 108, pp. 133–141, 2019.

[22] D. A. Van Valen, T. Kudo, K. M. Lane, D. N. Macklin, N. T. Quach, M. M. DeFelice, I. Maayan, Y. Tanouchi, E. A. Ashley, and M. W. Covert, "Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments," *PLoS computational biology*, vol. 12, no. 11, p. e1005177, 2016.

[23] M. S. Schwartz, E. Moen, G. Miller, T. Dougherty, E. Borba, R. Ding, W. Graf, E. Pao, and D. V. Valen, "Caliban: Accurate cell tracking and lineage construction in live-cell imaging experiments with deep learning," *bioRxiv*, 2023. [Online]. Available: https://www.biorxiv.org/content/early/2023/09/12/803205

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[25] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson, S. Buch, D. Card, R. Castellon, N. Chatterji, A. Chen, K. Creel, J. Q. Davis, D. Demszky, C. Donahue, M. Doumbouya, E. Durmus, S. Ermon, J. Etchemendy, K. Ethayarajh, L. Fei-Fei, C. Finn, T. Gale, L. Gillespie, K. Goel, N. Goodman, S. Grossman, N. Guha, T. Hashimoto, P. Henderson, J. Hewitt, D. E. Ho, J. Hong, K. Hsu, J. Huang, T. Icard, S. Jain, D. Jurafsky, P. Kalluri, S. Karamcheti, G. Keeling, F. Khani, O. Khattab, P. W. Koh, M. Krass, R. Krishna, R. Kuditipudi, A. Kumar, F. Ladhak, M. Lee, T. Lee, J. Leskovec, I. Levent, X. L. Li, X. Li, T. Ma, A. Malik, C. D. Manning, S. Mirchandani, E. Mitchell, Z. Munyikwa, S. Nair, A. Narayan, D. Narayanan, B. Newman, A. Nie, J. C. Niebles, H. Nilforoshan, J. Nyarko, G. Ogut, L. Orr, I. Papadimitriou, J. S. Park, C. Piech, E. Portelance, C. Potts, A. Raghunathan, R. Reich, H. Ren, F. Rong, Y. Roohani, C. Ruiz, J. Ryan, C. Ré, D. Sadigh, S. Sagawa, K. Santhanam, A. Shih, K. Srinivasan, A. Tamkin, R. Taori, A. W. Thomas, F. Tramèr, R. E. Wang, W. Wang, B. Wu, J. Wu, Y. Wu, S. M. Xie, M. Yasunaga, J. You, M. Zaharia, M. Zhang, T. Zhang, X. Zhang, Y. Zhang, L. Zheng, K. Zhou, and P. Liang, "On the opportunities and risks of foundation models," 2022.

[26] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[27] OpenAI, "Gpt-4 technical report," 2023.

[28] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli *et al.*, "Evolutionary-scale prediction of atomic-level protein structure with a language model," *Science*, vol. 379, no. 6637, pp. 1123–1130, 2023.

[29] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[30] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," 2021.

[31] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "Dinov2: Learning robust visual features without supervision," 2023.

[32] Y. Fang, W. Wang, B. Xie, Q. Sun, L. Wu, X. Wang, T. Huang, X. Wang, and Y. Cao, "Eva: Exploring the limits of masked visual representation learning at scale," 2022.

[33] 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.

[34] 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*, vol. 35, pp. 23 716–23 736, 2022.

[35] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[36] Y. Huang, X. Yang, L. Liu, H. Zhou, A. Chang, X. Zhou, R. Chen, J. Yu, J. Chen, C. Chen, H. Chi, X. Hu, D.-P. Fan, F. Dong, and D. Ni, "Segment anything model for medical images?" 2023.

[37] Y. Zhang, T. Zhou, S. Wang, P. Liang, and D. Z. Chen, "Input augmentation with sam: Boosting medical image segmentation with segmentation foundation model," 2023.

[38] W. Lei, X. Wei, X. Zhang, K. Li, and S. Zhang, "Medlsam: Localize and segment anything model for 3d medical images," 2023.

[39] P. Shi, J. Qiu, S. M. D. Abaxi, H. Wei, F. P.-W. Lo, and W. Yuan, "Generalist vision foundation models for medical imaging: A case study of segment anything model on zero-shot medical segmentation," *Diagnostics*, vol. 13, no. 11, p. 1947, 2023.

[40] M. Hu, Y. Li, and X. Yang, "Skinsam: Empowering skin cancer segmentation with segment anything model," 2023.

[41] R. Deng, C. Cui, Q. Liu, T. Yao, L. W. Remedios, S. Bao, B. A. Landman, L. E. Wheless, L. A. Coburn, K. T. Wilson, Y. Wang, S. Zhao, A. B. Fogo, H. Yang, Y. Tang, and Y. Huo, "Segment anything model (sam) for digital pathology: Assess zero-shot segmentation on whole slide imaging," 2023.

[42] F. Hörst, M. Rempe, L. Heine, C. Seibold, J. Keyl, G. Baldini, S. Ugurel, J. Siveke, B. Grünwald, J. Egger, and J. Kleesiek, "Cellvit: Vision transformers for precise cell segmentation and classification," 2023.

[43] A. Archit, S. Nair, N. Khalid, P. Hilt, V. Rajashekar, M. Freitag, S. Gupta, A. Dengel, S. Ahmed, and C. Pape, "Segment anything for microscopy," *bioRxiv*, 2023. [Online]. Available: https://www.biorxiv.org/content/early/2023/08/22/2023.08.21.554208

[44] C. Edlund, T. R. Jackson, N. Khalid, N. Bevan, T. Dale, A. Dengel, S. Ahmed, J. Trygg, and R. Sjögren, "Livecell—a large-scale dataset for label-free live cell segmentation," *Nature methods*, vol. 18, no. 9, pp. 1038–1045, 2021.

[45] C. Spahn, E. Gómez-de Mariscal, R. F. Laine, P. M. Pereira, L. von Chamier, M. Conduit, M. G. Pinho, G. Jacquemet, S. Holden, M. Heilemann *et al.*, "Deepbacs for multi-task bacterial image analysis using open-source deep learning approaches," *Communications Biology*, vol. 5, no. 1, p. 688, 2022.

[46] G. Mathieu, E. D. Bachir *et al.*, "Brifiseg: a deep learning-based method for semantic and instance segmentation of nuclei in brightfield images," *arXiv preprint arXiv:2211.03072*, 2022.

[47] K. J. Cutler, C. Stringer, P. A. Wiggins, and J. D. Mougous, "Omnipose: a high-precision morphology-independent solution for bacterial cell segmentation," *bioRxiv*, 2021.

[48] K. J. Cutler, C. Stringer, T. W. Lo, L. Rappez, N. Stroustrup, S. Brook Peterson, P. A. Wiggins, and J. D. Mougous, "Omnipose: a high-precision morphology-independent solution for bacterial cell segmentation," *Nature methods*, vol. 19, no. 11, pp. 1438–1448, 2022.

[49] H. Kim, J. Shin, E. Kim, H. Kim, S. Hwang, J. E. Shim, and I. Lee, "Yeastnet v3: a public database of data-specific and integrated functional gene networks for saccharomyces cerevisiae," *Nucleic acids research*, vol. 42, no. D1, pp. D731–D736, 2014.

[50] N. Dietler, M. Minder, V. Gligorovski, A. M. Economou, D. A. H. Lucien Joly, A. Sadeghi, C. H. Michael Chan, M. Koziński, M. Weigert, A.-F. Bitbol *et al.*, "Yeaz: A convolutional neural network for highly accurate, label-free segmentation of yeast microscopy images," *bioRxiv*, pp. 2020–05, 2020.

[51] J. C. Caicedo, A. Goodman, K. W. Karhohs, B. A. Cimini, J. Ackerman, M. Haghighi, C. Heng, T. Becker, M. Doan, C. McQuin *et al.*, "Nucleus segmentation across imaging experiments: the 2018 data science bowl," *Nature methods*, vol. 16, no. 12, pp. 1247–1253, 2019.

[52] Y. Li, H. Mao, R. Girshick, and K. He, "Exploring plain vision transformer backbones for object detection," in *European Conference on Computer Vision*. Springer, 2022, pp. 280–296.

[53] 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: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[54] Y. Wang, X. Zhang, T. Yang, and J. Sun, "Anchor detr: Query design for transformer-based detector," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, no. 3, 2022, pp. 2567–2575.

[55] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[56] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016.

[57] X. Zhang, X. Wang, G. Shivashankar, and C. Uhler, "Graph-based autoencoder integrates spatial transcriptomics with chromatin images and identifies joint biomarkers for alzheimer's disease," *Nature Communications*, vol. 13, no. 1, p. 7480, 2022.

[58] K. D. Yang, A. Belyaeva, S. Venkatachalapathy, K. Damodaran, A. Katcoff, A. Radhakrishnan, G. Shivashankar, and C. Uhler, "Multi-domain translation between single-cell imaging and sequencing data using autoencoders," *Nature communications*, vol. 12, no. 1, p. 31, 2021.

[59] S. Shah, E. Lubeck, W. Zhou, and L. Cai, "seqfish accurately detects transcripts in single cells and reveals robust spatial organization in the hippocampus," *Neuron*, vol. 94, no. 4, pp. 752–758, 2017.

[60] D. Dar, N. Dar, L. Cai, and D. K. Newman, "Spatial transcriptomics of planktonic and sessile bacterial populations at single-cell resolution," *Science*, vol. 373, no. 6556, p. eabi4882, 2021.

[61] K. M. Jablonka, Q. Ai, A. Al-Feghali, S. Badhwar, J. D. Bocarsly, A. M. Bran, S. Bringuier, L. C. Brinson, K. Choudhary, D. Circi *et al.*, "14 examples of how llms can transform materials science and chemistry: a reflection on a large language model hackathon," *Digital Discovery*, vol. 2, no. 5, pp. 1233–1250, 2023.

[62] E. Nguyen, M. Poli, M. Faizi, A. Thomas, C. Birch-Sykes, M. Wornow, A. Patel, C. Rabideau, S. Massaroli, Y. Bengio, S. Ermon, S. A. Baccus, and C. Ré, "Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution," 2023.

[63] D. Feldman, A. Singh, J. L. Schmid-Burgk, R. J. Carlson, A. Mezger, A. J. Garrity, F. Zhang, and P. C. Blainey, "Optical pooled screens in human cells," *Cell*, vol. 179, no. 3, pp. 787–799, 2019.

[64] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld, "Adaptive histogram equalization and its variations," *Computer vision, graphics, and image processing*, vol. 39, no. 3, pp. 355–368, 1987.

[65] 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*, vol. 2, p. e453, 6 2014. [Online]. Available: https://doi.org/10.7717/peerj.453

[66] J. Hosang, R. Benenson, and B. Schiele, "Learning non-maximum suppression," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4507–4515.

[67] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[68] 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: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 2020, pp. 213–229.

[69] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[70] 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 32*. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[71] W. Falcon and The PyTorch Lightning team, "PyTorch Lightning," Mar. 2019. [Online]. Available: https://github.com/Lightning-AI/lightning

Declarations

Code Availability

The Van Valen lab is a strong believer in open-source software development. A github repository will be made available at https//github.com/vanvalenlab as soon as a library with satisfactory software engineering standards is compiled.

Data Availability

The dataset used to develop CellSAM is available at https://deepcell.readthedocs.io/en/master/data-gallery/index.html for non-profit use.

Author Contributions

UI, MM, YY, and DVV conceived the project; UI, MM, QL, YY, and DVV performed algorithm design for CellFinder and CellSAM; MM implemented the CellSAM architecture; UI, MM, and QL implemented CellFinder. UI and MM carried out the experiments and evaluations of the method. GG and PP provided input for developing CellFinder; QL and UI performed model benchmarking; QL and RD developed data pipelines, RD developed the computational infrastructure for model training; RD, EP, EP, MS, QL, and RB performed data engineering; SL, APG, RD, and RB performed the DeepCell Label-CellSAM integration, RB and DVV supervised the software engineering, DVV supervised the project.

Acknowledgements

We thank Leeat Keren, Noah Greenwald, Sam Cooper, Jan Funke, Uri Manor, Joe Horsman, Michael Baym, Paul Blainey, Ian Cheeseman, Manuel Leonetti, Changhua Yu, Neehar Kondapaneni, and Elijah Cole for valuable conversations and insightful feedback. We also thank William Graf, Geneva Miller, and Kevin Yu, whose time in the Van Valen lab established the infrastructure and software tools that made this work possible. We thank Nader Khalil, Alec Fong, and the entire Brev.dev team for their support in establishing the computational infrastructure required for this work. We utilized images of the HeLa cell line in this research. Henrietta Lacks and the HeLa cell line established from her tumor cells without her knowledge or consent in 1951 has significantly contributed to scientific progress and advances in human health. We are grateful to Lacks, now deceased, and the Lacks family for their contributions to biomedical research. This work was supported by awards from the Shurl and Kay Curci Foundation (to DVV), the Rita Allen Foundation (to DVV), the Susan E. Riley Foundation (to DVV), the Pew-Stewart Cancer Scholars program (to DVV), the Gordon and Betty Moore Foundation (to DVV), the Schmidt Academy for Software Engineering (to SL), the Michael J. Fox Foundation through the Aligning Science Across Parkinson's consortium (to DVV), the Heritage Medical Research Institute (to DVV), the National Institutes of Health New Innovator program (DP2-GM149556) (to DVV), the National Institutes of Health HuBMAP consortium (OT2-OD033756) (to DVV), and the Howard Hughes Medical Institute Freeman Hrabowski Scholars program (to DVV). National Institutes of Health (R01-MH123612A) (to PP). NIH/Ohio State University (R01-DC014498) (to PP). Chen Institute (to PP). The Emerald Foundation and Black in Cancer (to UI). Caltech Presidential Postdoctoral Fellowship Program (PPFP) (to UI).

Disclosures

David Van Valen is a co-founder and Chief Scientist of Barrier Biosciences and holds equity in the company. All other authors declare no competing interests.

A Dataset Construction

To train CellSAM, we combined nine separate datasets spanning a variety of modalities: TissueNet[18], DeepBacs[45], BriFiSeg[46], Cellpose[15,16], Omnipose[47,48], YeastNet[49], YeaZ[50], the 2018 Kaggle Data Science Bowl (DSB)[51], and an internally collected dataset of phase microscopy images across eight mammalian cell lines (Phase400). The LIVECell[44] dataset was held out for zero-shot testing. Our collective dataset included images across multiple imaging modalities (brightfield, phase contrast, h&e staining, fluorescence, and mass cytometry), imaging targets (histology sections, yeast, cell culture, bacteria, nuclei), length scales, and morphologies. During preprocessing, every image in our dataset was normalized using Contrast Limited Adaptive Histogram Equalization (CLAHE)[64] with a kernel size of 128 pixels. We treated nuclear and whole-cell channels as green and blue channels in an RGB image, respectively, and the red channel is always blank. We moved the green channel to blue for nuclear-only datasets (i.e., BriFiSeg and DSB) to keep the blue channel always non-empty.

If available, we used pre-determined train/val/test splits for each dataset; otherwise, we introduced 80-10-10% data splits. For datasets with multiple fields of view of the same object set, we required all FOVs to belong to the same split. We defer all duplicated samples to the train split for published datasets with a pre-existing data leak. Our assembled dataset uses a fixed image size of 512 by 512 pixels. Images shorter than 512 pixels on either axis are zero-padded up to 512. For images with more than 512 pixels on either axis, we tiled them to 512 by 512 pixels with a 25% overlap and filled the empty regions with zeros. Any cropped images without valid annotations were removed. We follow a widely used annotation scheme for labeling our masks, with zero representing the background and unique positive integers representing different objects. While this format precludes accurate segmentation of overlapping objects, labels of this kind were not present in the dataset we compiled. We filtered out invalid cell labels if 1) the label contained disjoint regions, typically caused by random mouse clicking; 2) the label has only a 1-pixel height or width. The cropped images with filtered annotations are used for training, validation, and testing. LIVECell[44] annotations were converted from the COCO format to this labeling format for consistency. We used cellpose's[16] pre-processing function `livecell_ann_to_masks()` to remove overlapping regions. To match the phase contrast cell size in the training set, we rescale the LIVECell images by 2.0 before the standard image preprocessing pipeline. We use scikit-image[65] `skimage.transform.rescale()` function with `bicubic` interpolation for images and `nearest` interpolation for annotation masks.

To summarize our dataset format, our images have RGB channels with a fixed size of 512 by 512 pixels, stored in shape (3, 512, 512) float32 array in the range [0, 1]. The blue channel was the main channel, reserved for whole-cell images, and always non-empty. The green channel was the supplementary channel used for nuclear images but could be empty. The blue channel was used if only nuclear images were available (e.g., for the DSB dataset). The red channel was always empty. Our label masks had the same height and width as the images, stored with the shape (1, 512, 512) int32 in the range [0, number of objects]. We stored the processed dataset in two formats. The numpy npy format was used for CellSAM fine-tuning and model evaluation. The COCO format[53] was used for CellFinder and ViT backbone training.

B CellSAM Architecture.

We adapted Anchor DETR[54] for the object detector for CellSAM (CellFinder). This choice was motivated by Anchor DETR being non-maximum suppression (NMS)[66] free. NMS suppresses bounding boxes with a high amount of overlap to remove duplicate detections. While this works well for natural images, cellular images often have tightly clustered objects, and NMS-based methods such as the R-CNN family[55,56] can suffer from a low recall in this setting. We replaced the Anchor DETR's ResNet[67] backbone with the vision transformer (ViT)[29] from the SAM model[35]; specifically, we used the base-sized ViT (ViT-B).

As the maximum number of cells per image is generally no more than 1000, we increased the number of queries q to 3500, 3.5 times the maximum number of cells, based on Fig. 12 in DETR[68], which provides an estimate of the number of queries needed for a DETR method to detect all objects. We used only one pattern p for the Anchor generation as most objects in cellular detection are usually of similar scale.

Training CellFinder We used a base learning rate of 10^{-4} for the Anchor DETR head and 10^{-5} for the SAM-ViT backbone. We use weight decay of 10^{-4} and clip norm of 0.1. We apply a dropout of 0.1. We use AdamW[69] with a step-wise learning rate scheduler that drops the learning rate by 10% after 70% of the epochs. We train CellFinder for 500 epochs (1000 for smaller datasets) with a batch size of 2 across 16 GPUs.

Finetuning CellSAM After we trained CellFinder with the SAM-ViT backbone, the SAM-ViT output features were no longer aligned with the rest of the model (i.e., the prompt encoder and mask decoder). To close this distribution gap, we froze the SAM-ViT (such that it continues to function well with CellFinder) and trained the neck of the SAM model. The neck is a 2D-convolutional neural network that embeds the ViT features (e.g., 768 for SAM-ViT-B) to a 256-dimensional embedding that is then used as the primary feature vector for the rest of the model (prompt embedding and mask decoder). We trained this neck using ground-truth bounding boxes as inputs and segmentation masks of individual cells as targets. We used a learning rate of 10^{-4} and weight decay of 10^{-4} for this training. We also used AdamW[69] for this training and did not clip the gradient.

B.1 Inference

At inference, we followed the following workflow. First, the input was passed through the Anchor DETR fine-tuned ViT-B. This resulted in an embedding dimension of 768. This embedding was then passed as an input to two parts of CellSAM, 1) the trained Anchor DETR module (CellFinder) and 2) the fine-tuned neck, which is a 2D convolutional network reducing the embedding dimensionality further to 256. The bounding box outputs of CellFinder were then sent into the prompt encoder, resulting in the prompt embedding. The prompt embeddings and neck embedding were then passed to the mask decoder, which outputs pixel-wise probabilities for the cell and another IoU-based confidence value for the prediction as a whole. This results in a tensor of shape $N \times W \times H$, where N corresponds to the number of cells predicted. This tensor was processed with a sigmoid and a threshold operation, resulting in binarized images. Depending on the metric used, we either use this tensor directly together with the N scores (specifically for computation of the

coco AP @ 0.5 IoU) or we compute the argmax over the cell dimension N to generate a tensor $W \times H$, where each pixel corresponds to an integer that is unique for each cell.

Thresholding. Given CellSAM's model architecture, we have three different thresholds at inference time. First, we had a threshold on the bounding boxes generated by CellFinder, which we set to 0.4 across all datasets. After the boxes were passed through the Mask Decoder, we had an overall mask score outputted by the IoU prediction head of the Mask Decoder, which we set to 0.5. Lastly, we thresholded the mask decoder output after applying the sigmoid function to each pixel, which we set at 0.5.

CellSAM Postprocessing. We use the same postprocessing steps that are used by SAM[35]. This consisted of hole filling and island removal for each predicted cell.

B.1.1 Model Implementation and Training

CellSAM is implemented in pytorch[70]. For CellFinder we modify the official Anchor DETR repo[1]. For CellSAM, we modify the official Segment Anything repo[2]. We use pytorch lightning[71] to scale the training. Prototyping was done using NVIDIA's RTX 4090. We used machines with either NVIDIA A6000s or A100s (40GB and 80GB versions) for the experiments in the paper.

C Benchmarking

We benchmarked the performance of CellSAM models against Cellpose[15,16] trained on our compiled datasets.

C.1 Cellpose Model Training.

We follow the hyper-parameters described in the original paper[16] to train specialist and generalist Cellpose models from scratch. We use the SGD optimizer with a weight decay of 10^{-4} and a batch size of eight. We train each model for 300 epochs with a base learning rate of 0.1. The learning rate increases linearly from 0 to 0.1 over the first ten epochs, then decreases by a factor of two every five epochs after the 250th epoch. The main channel (`--chan`) is 3 (blue), and the supplementary channel (`--chan2`) is 2 (green). Other hyper-parameters were kept at the default setting. We trained each model on a single NVIDIA A6000 GPU with 11.4GB GPU Memory utilization. In total, we train nine specialist models and one generalist model.

C.2 Metrics

We used the Metrics package present in the DeepCell library[18,23], which is a set of tools for object-level evaluation of cell segmentations. Predictions that match the ground truth labels (determined by a mask IoU ≥ 0.6) are true positives (TP), predictions with no matching ground truth labels are false positives (FP), and ground truth labels without a valid match are false negatives. We compute the recall, precision, and F1 scores using the following formulas:

- Recall: recall $= \frac{\text{TP}}{\text{TP+FN}}$.
- Precision: precision $= \frac{\text{TP}}{\text{TP+FP}}$.

[1] https://github.com/megvii-research/AnchorDETR
[2] https://github.com/facebookresearch/segment-anything

- F1: $F_1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$.

Details of the implementation of these metrics are described in prior work[23].

We also used the COCO evaluation metrics[53] during CellFinder's development. The COCO metrics are a widely used benchmark for assessing the object-level quality of object detection and instance segmentation methods. These metrics report Average Precision (AP), the area under the Precision-Recall curve for a given object class. In our case, we only had a single object class - cells. The AP is computed for different IoU thresholds, ranging from 0.5 to 0.95, with a step size of 0.05. We report the mean AP across all IoU thresholds, denoted as `mAP`, as well as the AP at IoU=0.5, denoted as `AP50`, to quantify CellFinder's performance. Because the object density is much higher in cellular images than in natural images, we modified the limit for the maximum number of detections from 100 to 10,000. We also fed the actual confidence score per binary prediction of the CellSAM model to the COCO evaluator. For the Cellpose models, we used a fixed confidence score of 1.0.

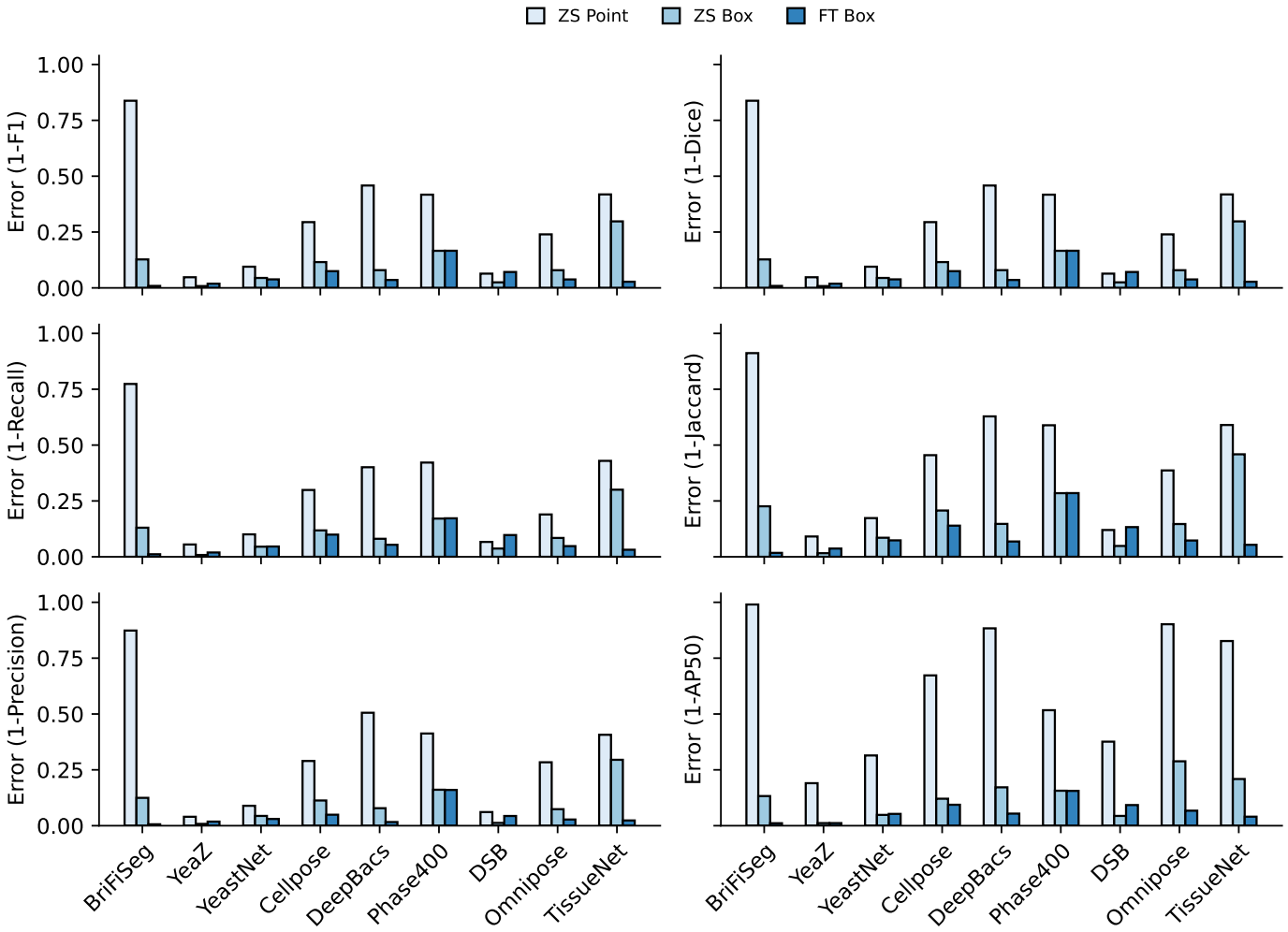

Fig. S1 Per dataset performance comparing zero-shot point prompting, zero-shot box prompting, and fine-tuned box prompting across a suite of metrics from the DeepCell package, and additionally, we included the AP50 from the COCO metrics. We show the error rate (1-metric) on these bar plots. We demonstrate CellSAM-specificand CellSAM-generalsuperior performance across multiple datasets and multiple evaluation metrics.

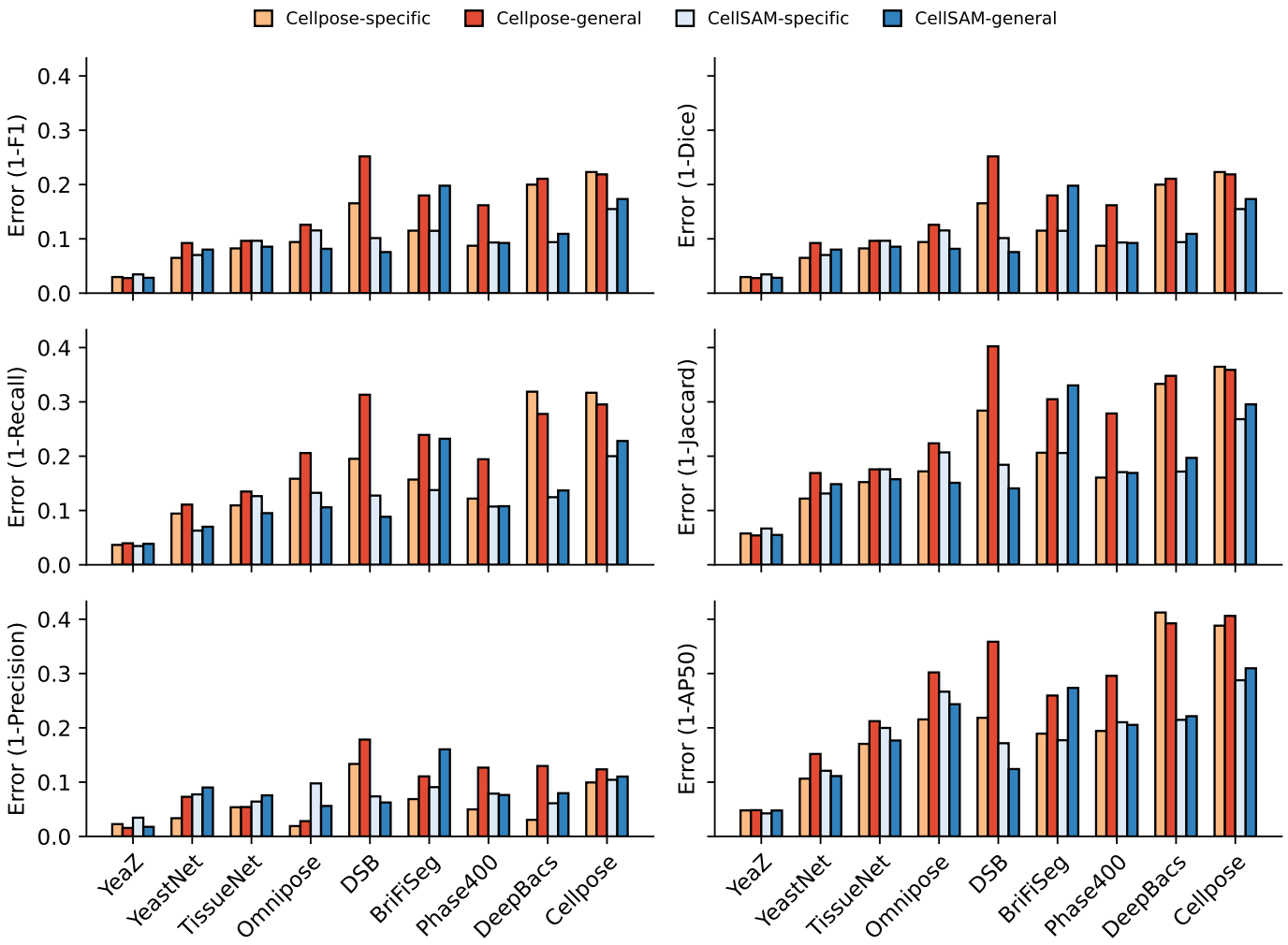

Fig. S2 Per dataset performance across a suite of metrics from the DeepCell package, and additionally, we included the AP50 from the COCO metrics. We show the error rate (1-metric) on these bar plots. We demonstrate CellSAM-specificand CellSAM-generalsuperior performance across multiple datasets and evaluation metrics.