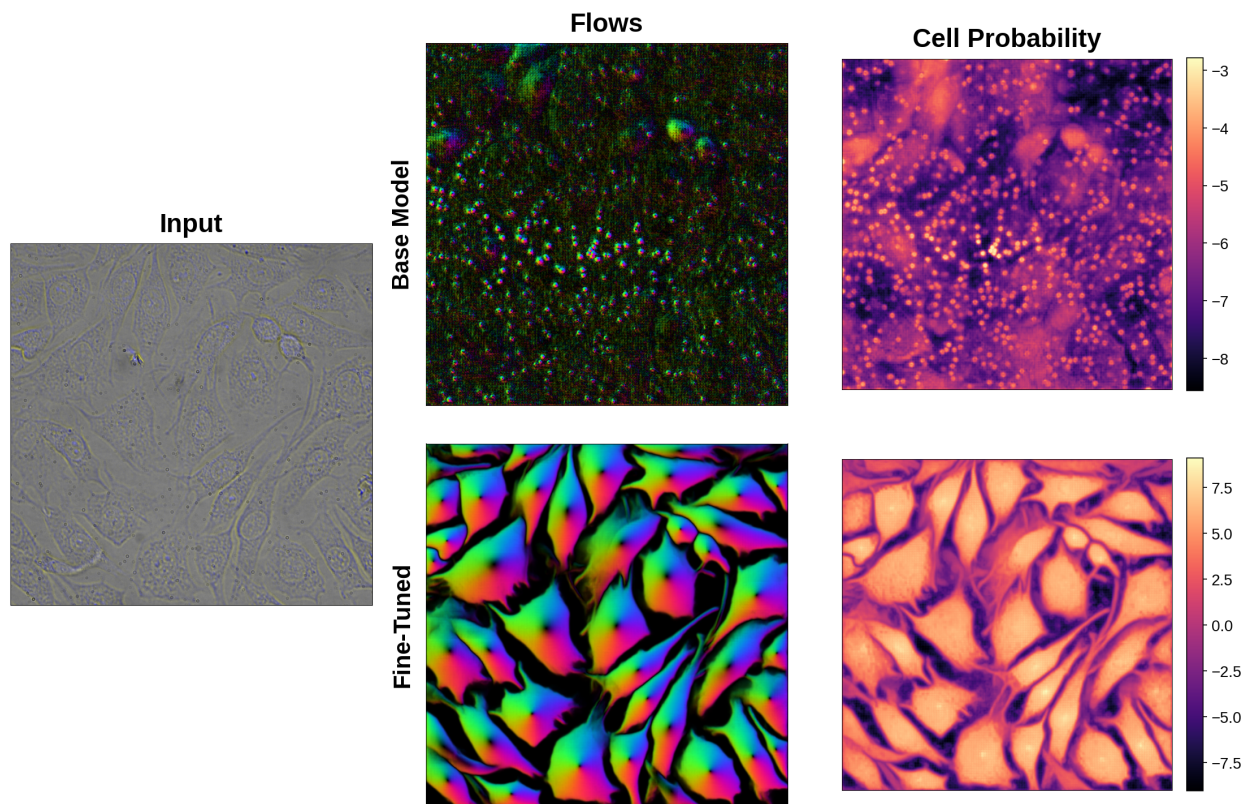




Adapting Without Forgetting

A Benchmark and Parameter-Efficient Fine-Tuning Study of Vision Foundation Models for Cell Segmentation in Microscopy

Max Haraldsson
Anton Widengård

Master's thesis in Computer science and engineering

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026

MASTER'S THESIS 2026

Adapting Without Forgetting

A Benchmark and Parameter-Efficient Fine-Tuning Study of Vision
Foundation Models for Cell Segmentation in Microscopy

Max Haraldsson & Anton Widengård



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026

Adapting Without Forgetting
A Benchmark and Parameter-Efficient Fine-Tuning Study of Vision Foundation
Models for Cell Segmentation in Microscopy
Max Haraldsson & Anton Widengård
© Max Haraldsson & Anton Widengård, 2026
Supervisor: Jesús Pineda
Advisor: IFLAI AB
Examiner: Jennifer Alvé
Master's Thesis 2026
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg

Cover: Segmentation masks produced by the fine-tuned Cellpose-SAM model on a
brightfield microscopy image. Typeset in L^AT_EX
Gothenburg, Sweden 2026

Abstract

Automated cell segmentation in microscopy images is a critical task in pharmaceutical research and biological discovery. Existing supervised approaches require extensive annotated data and must be retrained for each new cell type or imaging condition. Foundation models offer a promising alternative through their ability to generalise across diverse domains without retraining. However, a systematic comparison of their generalisation across diverse microscopy conditions has been lacking. Moreover, adapting such models to specific datasets through fine-tuning risks catastrophic forgetting, potentially undermining the very generalisation that makes them valuable.

This thesis benchmarks two state-of-the-art foundation models for cell segmentation, CellSAM and Cellpose-SAM, across several datasets spanning multiple imaging modalities. Cellpose-SAM consistently outperformed CellSAM, particularly at stricter evaluation thresholds, attributed to its flow-based instance separation, which handles touching and overlapping cells more effectively than CellSAM’s prompt-based approach.

Cellpose-SAM was subsequently fine-tuned using QLoRA, a parameter-efficient method that adapts only a small fraction of the model’s weights. Fine-tuning on a small curated dataset yielded substantial improvement on challenging microscopy images while retaining near-baseline performance on held-out general data, demonstrating that catastrophic forgetting can be avoided with careful, parameter-efficient fine-tuning. The entire training process required approximately 15 minutes on a single A100 GPU, illustrating that meaningful adaptation of large vision foundation models is achievable with modest computational resources.

Keywords: cell segmentation, foundation models, microscopy, Cellpose-SAM, CellSAM, parameter-efficient fine-tuning, QLoRA, LoRA, catastrophic forgetting, instance segmentation, vision transformers, feature distillation

Acknowledgements

We want to thank our supervisor, Jesús Pineda, for the support across the semester, and the team at IFLAI for making this thesis possible. We also want to thank our examiner, Jennifer Alvéén, for continuously supporting us throughout the project. The computations were enabled by the supercomputing resource Berzelius provided by National Supercomputer Centre at Linköping University and the Knut and Alice Wallenberg foundation. Finally, we want to thank the Soft Matter Lab for their feedback and interesting discussions.

Max Haraldsson & Anton Widengård, Gothenburg, 2026-06-02

Contents

List of Figures	xi
List of Tables	xv
List of Abbreviations	xvii
1 Introduction	1
1.1 Automatic cell segmentation	1
1.2 Research Gap	2
1.3 Aim and objective	2
1.4 Limitations	3
2 Preliminaries	5
2.1 Imaging Modalities	5
2.2 Vision transformers	7
2.3 Evaluation Metrics	8
3 Benchmarking	11
3.1 Segmentation models	11
3.1.1 SAM	11
3.1.2 CellSAM	11
3.1.3 Cellpose-SAM	12
3.1.4 StarDist	13
3.2 Method	13
3.2.1 Datasets	14
3.2.2 Benchmark procedure	15
3.3 Results	16
3.3.1 Model comparison	16
3.3.2 Failure cases	19
3.4 Discussion	20
4 Fine-tuning	23
4.1 Fine-tuning techniques	23
4.1.1 LoRA	23
4.1.2 QLoRA	24
4.1.3 Feature Distillation	24

4.2	Fine-tuning methodology	25
4.2.1	Dataset	25
4.2.2	Model architecture and tuning strategies	26
4.2.3	Implementation details and training strategy	27
4.2.4	Fine-tuning evaluation	28
4.3	Results	29
4.3.1	Hyperparameter optimisation	29
4.3.2	Model selection	30
4.3.3	Domain adaptation (Held-out test set)	33
4.3.4	Out-of-distribution generalisation (CTC: Fluo-C2DL-MS)	38
4.3.5	Domain retention (Cellpose test set)	39
4.3.6	Further generalisation evaluation	40
4.4	Discussion	43
4.4.1	Fine-tuning strategy and hyperparameter optimisation	43
4.4.2	Qualitative analysis of fine-tuning result	44
4.4.3	Catastrophic forgetting	48
5	Conclusion	49
5.1	Benchmark	49
5.2	Fine-tuning	49
6	Future work	51
	Bibliography	53
A	LoRA equations	I
B	Benchmark Results	III
C	Parameter Sweep Results	VII
D	Further evaluation results	XIX
E	Qualitative analysis	XXI
E.1	ALFI	XXIV
E.2	Phase400 CellSAM	XXIV

List of Figures

1.1	Project outline of the benchmarking and fine-tuning stage. The three models, Cellpose-SAM, CellSAM, and StarDist, were evaluated on 18 datasets comprising 4 different modalities. The best performing of these models was then fine-tuned and evaluated against the original model based on three different datasets.	3
2.1	Examples of different microscopy modalities: fluorescence microscopy (FLU0-N2DH-SIM+), Phase contrast (PHC-C2DL-PSC), and differential interference contrast (DIC-C2DH-HELA) imaging, all sourced from the Cell Tracking Challenge (CTC) [2]; and brightfield microscopy (BF) from the EVICAN dataset. [13]	6
2.2	Overview of DIC microscopy. Polarised light enters a prism, which splits it into two separated, orthogonally polarised beams. The beams traverse slightly different optical paths through the sample. A second prism recombines them, and the analyser causes interference. The resulting intensity variation reflects the optical path length gradient across the sample. Modified from Heydarian et al. [17] under a CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).	7
3.1	Overview of the CellSAM architecture where an input image is sent through a fine-tuned SAM ViT backbone. The features are then sent into both CellFinder, a modified AnchorDETR that detects and creates bounding boxes of cells, which are then sent into the SAM decoder as prompts together with the backbone features. The decoder then generates segmentation masks of cells found in the prompted bounding boxes. Image reprinted from M Marks et al. [6] under a CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).	12
3.2	Illustration of the Cellpose-SAM segmentation. From left to right: the input image, the predicted flow field, and the cell probability map (high probability shown in yellow, low probability in dark blue/purple)	13
3.3	Number of images from each dataset used in the benchmarking.	15
3.4	Comparison of mean segmentation performance across all benchmarking datasets. Cellpose-SAM outperforms both CellSAM and StarDist across all metrics.	16

3.5	Mean instance IoU and AP@0.5 across all benchmarking datasets, split per imaging modality. Cellpose-SAM outperforms both CellSAM and StarDist with the exception on fluorescence images where CellSAM exhibits the best performance by a small margin.	17
3.6	Comparison of segmentation accuracy between Cellpose-SAM (left) and CellSAM (right) on an image from the Fluo-N2DH-GOWT1 dataset from the CTC, with FP in red, FN in green, and TP in yellow. Cellpose-SAM shows a superior segmentation accuracy compared to CellSAM, which finds all cells but overpredicts to a greater extent. . .	18
3.7	Comparison of the segmentation performance of Cellpose-SAM without contrast enhancement, with Gaussian Blur, and with Contrast Limited Adaptive Histogram Equalization (CLAHE). Evaluated on three different images with low contrast and uneven lighting. True positives (TP) are shown in yellow, false positives (FP) in red, and false negatives (FN) in green. Significant improvement can be seen by applying Gaussian Blur on images with uneven lighting and CLAHE on images with low contrast.	19
3.8	Comparison of Cellpose-SAM segmentation performance on an image from the dataset Revvity without manually specifying the diameter parameter (top row), and with a manually set diameter (bottom row). Without setting a diameter, the cells are too large, resulting in the model predicting background noise instead. But by specifying the diameter, the model finds and segments the majority of the cell population.	20
4.1	Number of images from each dataset included in the curated fine-tuning dataset.	25
4.2	Comparison of segmentation performance of the base model and the five best performing configurations on the fine-tuning test set, Cellpose dataset, and Fluo-C2DL-MSD dataset from the CTC, measured by iIoU and AP@0.5.	31
4.3	Comparison of segmentation performance measured by AP@0.5 and instance IoU between the base Cellpose-SAM and the final fine-tuned model across the fine-tuning test set, Cellpose dataset, and the Fluo-C2DL-MSD from the CTC.	33
4.4	Segmentation performance of the base and fine-tuned Cellpose-SAM model on the fine-tuning test set across all six metrics. Shaded columns indicate lower-is-better metrics.	34
4.5	Comparison of segmentation performance between the Fine-tuned model and the base model, per dataset in the fine-tuning test set. Sorted by the highest performance difference from left to right. The upper graph shows the difference in instance IoU performance, and the lower, AP@0.5.	35

4.6	Performance of the base- (center) and fine-tuned (right) model on the Revvity image containing cells with an out-of-distribution size. Following fine-tuning, the model, compared to the base model, can find and segment cells where true positives are shown in yellow, false positives in red, and false negatives in green.	36
4.7	Input image, base- and fine-tuned-model segmentation on an image from the dataset PhC-C2DH with true positives in yellow, false positives in red, and false negatives in green. It can be seen that the fine-tuned model finds all cells but also oversegments in its prediction.	36
4.8	Input image, base- and fine-tuned-model segmentation on a DIC-C2DH image with true positives in yellow, false positives in red, and false negatives in green. The fine-tuned model can, compared to the base model, more accurately distinguish between the background and foreground, resulting in more precise segmentations.	37
4.9	Input image, base- and fine-tuned-model segmentation on a Fluo-N2DH-SIM image with true positives in yellow, false positives in red, and false negatives in green.	37
4.10	Input image, base- and fine-tuned-model segmentation on an image containing uneven lighting from the DIC-HepG2 dataset with true positives in yellow, false positives in red, and false negatives in green. As Cellpose-SAM does not include any contrast enhancement, the base model had a poor performance, but by fine-tuning, the performance, while not perfect, significantly increases.	38
4.11	Segmentation performance on the unseen out-of-distribution CTC dataset (Fluo-C2DL-MS , $n = 51$), evaluating generalisation beyond the fine-tuning distribution. Shaded columns indicate lower-is-better metrics.	39
4.12	Segmentation performance of base- and fine-tuned model on the Cellpose test set, used to monitor domain retention. Shaded columns indicate lower-is-better metrics.	40
4.13	Comparison of segmentation mask, gradient flows, and cell probability map for base- (upper) and fine-tuned (lower) model on an image from the ALFI dataset. While the image's ground truth mask only contains nuclei, making metric comparison infeasible, one can see in the figure that the fine-tuned model is more successful in finding the cells compared to the base model. In the predicted masks, true positives are shown in yellow, false positives in red, and false negatives in green.	41

4.14	Comparison of segmentation mask, flows, and cell probability between base- (upper) and fine-tuned model (lower) on an image from the StarDist_BF dataset. As the image contains a background of cells not included in the ground truth mask, direct metric comparison does not give any important insight. However, one can see that the fine-tuned model creates more accurate segmentations of these background cells, where it can, to a higher degree, distinguish actual background from the background cells. In the predicted mask, true positives are shown in yellow, false positives in red, and false negatives in green.	42
4.15	Flow and cell probability for base (upper) and fine-tuned (lower) model on an image from Revvity containing large out-of-distribution cells. While the base model simply finds background noise, the fine-tuned model can correctly find and segment these large cells.	45
4.16	Flow and cell probability for base (upper) and fine-tuned (lower) model on an image from the dataset PhC-C2DH . The fine-tuned model produces more accurate flow gradients and detects more details, as seen in the cell in the upper right corner.	46
4.17	Flow and cell probability for base (upper) and fine-tuned (lower) model on an image containing uneven lighting from the DIC-HepG2 dataset. While the base model only finds the solitary cell in the left part of the image, the fine-tuned model finds the majority of cells illustrating the effect of fine-tuning.	47
C.1	Comparison of performance across different Learning rates varying from $5e-4$ to $1e-6$. Fine-tuned configurations are shown as colored lines, and the original base model is shown in dashed black.	XIII
C.2	Comparison of performance when injecting QLoRA into the first half of blocks (blk 0-11), last half (blk 12-23), final blocks (18-23), and all blocks (0-23). A configuration of (rank 32, $\alpha = 64$, LR $5e-5$, batch size 8) was used with the exception of the green configuration, which used a rank of 64, and an $\alpha = 128$	XIV
C.3	Comparison of performance with varying QLoRA parameters (rank (16-32), α (32-128), and batch-size (1-8). blocks 12-23, LR $5e-5$. . .	XV
C.4	Comparison of performance when combining the best-performing configurations from previous sweeps.	XVI
C.5	Comparison of performance utilising feature distillation with configurations mirroring those in the top configuration sweep C.4.	XVII
E.1	The 5 images from the fine-tuning test-set images with the highest iIoU-gain	XXII
E.2	The 5 images from the fine-tuning test-set with the lowest iIoU-gain .	XXIII
E.3	Segmentation mask, gradient flows and cell probability map for ALFI sample	XXIV
E.4	Segmentation mask, gradient flows and cell probability map for sample from Phase400 dataset	XXIV

List of Tables

4.1	Distribution of normal and hard cases across the 80/10/10 dataset split. The discrepancies between the validation and test sets are the result of video sequence separation.	26
4.2	Metrics of the base model and the five best performing configurations across the fine-tuning test Set, Cellpose dataset, and the Fluo-C2DL-MS dataset from the CTC.	32
4.3	Segmentation performance on the fine-tuning test set. Best result per metric in bold	34
4.4	Segmentation performance of base- and fine-tuned model on unseen out-of-distribution (OOD) data (CTC: Fluo-C2DL-MS , $n = 51$). Best result per metric in bold	38
4.5	Segmentation performance on the Cellpose test set. Best result per metric in bold	39
B.1	Mean segmentation metrics by model across all datasets, best result per metric in bold	III
B.2	Benchmark segmentation results across datasets and modalities. Best result per dataset per metric shown in bold , with N indicating the number of images used from the dataset.	IV
C.1	Learning rate sweep results (blocks 12–23, rank 32, $\alpha = 64$, batch size 8). Best result per metric in bold	VIII
C.2	Block injection sweep results (rank 32, $\alpha = 64$, LR 4×10^{-5} , batch size 8). Best result per metric in bold	IX
C.3	Batch size and rank sweep results (blocks 12–23, LR 1×10^{-4}). Best result per metric in bold	X
C.4	Final combination sweep results (batch size 8). Best result per metric in bold	XI
C.5	Distillation sweep results (batch size 8). Configurations mirror those in the final combination sweep. Best result per metric in bold	XII
D.1	Comparison of performance between base-model (Base), and final fine-tuned model (FT) on the ALFI test set containing 80 DIC images with GT masks containing Nuclei.	XIX

D.2	Comparison of performance between base-model (Base), and final fine-tuned model (FT) on the Dataset Phase400 from CellSAM containing 54 PHC images.	XIX
D.3	Comparison of performance between base-model (Base), and final fine-tuned model (FT) on the StarDistBF dataset containing 27 BF images.	XX

List of Abbreviations

AP	Average Precision
BF	Brightfield
CTC	Cell Tracking Challenge
DIC	Differential Interference Contrast
EMA	Exponential Moving Average
FM	Foundation Model
FN	False Negative
FP	False Positive
HCS	High-Content Screening
IoU	Intersection over Union
iIoU	Instance Intersection over Union
LoRA	Low-Rank Adaptation
MLP	Multilayer Perceptron
MSA	Multi-Head Self-Attention
NF4	4-bit NormalFloat
NMS	Non-Maximum Suppression
OOD	Out-of-Distribution
PEFT	Parameter-Efficient Fine-Tuning
PHC	Phase Contrast
QLoRA	Quantized Low-Rank Adaptation
SAM	Segment Anything Model
SOTA	State of the Art
TP	True Positive
ViT	Vision Transformer

1

Introduction

At the core of modern pharmaceutical research and biological discovery lies the need to extract quantitative information from large numbers of microscopy images [1]. During the past decade, the field has been dominated by supervised deep learning architectures, such as U-Nets [2][3], which require extensive amounts of manually annotated data. These models typically lack generalisability, necessitating retraining for every new cell type or imaging condition [4].

To address these limitations, there is currently a shift towards Foundation Models (FMs), [5]. These models are trained on massive, diverse datasets, which allows them to generalise to new objects and conditions without re-training. However, most current FMs are trained on natural images and thus lack an understanding of biological structures like cells. Recent studies have investigated adapting FMs for segmentation to the microscopy domain [6][7], but a systematic evaluation of their generalisation capabilities across various imaging conditions and modalities remains limited.

This thesis, conducted in collaboration with IFLAI, a company specialising in AI for life sciences, addresses two interconnected challenges. First, it evaluates the zero-shot segmentation performance of state-of-the-art FMs across diverse microscopy conditions. Secondly, it develops a parameter-efficient fine-tuning strategy that improves domain-specific performance without catastrophic forgetting [8][9].

1.1 Automatic cell segmentation

Automated analysis of microscopy images is a fundamental part of modern pharmaceutical research and biological discovery. In drug development, high-content screening (HCS) workflows use automated microscopes to carry out multiparametric assays in an unbiased and quantitative fashion across all stages of the drug discovery pipeline [10]. Extracting meaningful measurements from these images, such as cell count, morphology, and organisation, requires that individual cells be outlined, a process known as instance segmentation. Despite its importance, accurate segmentation remains a bottleneck for many researchers [11]. Manual annotation is slow, subjective, and entirely infeasible at the scale of modern high-throughput experiments. Automated methods are therefore essential to efficiently segment images at scale.

The core challenge is the absence of a single standardised imaging protocol. Cells are imaged under widely varying conditions, spanning different modalities, cell types, magnifications, and staining protocols, and their appearance varies across these settings. Existing supervised methods are often tailored to specific modalities or require manual adjustments of hyperparameters for each experimental setting, resulting in poor generalisation across imaging platforms [12]. The annotation required to train such models is itself a bottleneck, as producing high-quality ground-truth masks demands expert biological knowledge and is time-consuming [11].

1.2 Research Gap

While the emergence of microscopy Foundation Models (FMs) such as CellSAM and Cellpose-SAM has reduced the dependency on scratch-built supervised models, there are still gaps that remain unaddressed. The models’ existing evaluations are mainly performed on their own curated datasets, which masks potential failure cases in complete zero-shot scenarios. Furthermore, there is a trade-off between specialization and generalisation. While fine-tuning is necessary to improve their performance, the current literature lacks a systematic exploration of parameter-efficient fine-tuning (PEFT). Consequently, it remains unclear how to optimize these large-scale models for microscopy tasks without inducing catastrophic forgetting or requiring extensive computational resources.

1.3 Aim and objective

The aim of this project is to evaluate and adapt FMs for cell segmentation in microscopy images. The work is conducted in collaboration with IFLAI, a company specialising in AI for life sciences, and is structured around two progressive objectives, shown in Figure 1.1. The primary objective is a systematic benchmark evaluation of state-of-the-art FMs for cell segmentation, specifically CellSAM and Cellpose-SAM, across a diverse set of publicly available microscopy datasets spanning multiple imaging modalities. This establishes which model generalises best in zero-shot settings and identifies the conditions under which each model fails.

The second objective involves fine-tuning the superior model on a dataset of challenging cases, with an explicit focus on parameter-efficient adaptation that preserves generalisation. Given the large scale of the encoder and the limited size of available annotated data, full fine-tuning risks catastrophic forgetting of broadly useful features. QLoRA is therefore employed to constrain adaptation to a small subset of parameters, and knowledge distillation is investigated as an additional regularisation strategy. The goal is to improve performance on the target domain without degrading performance on unseen data.

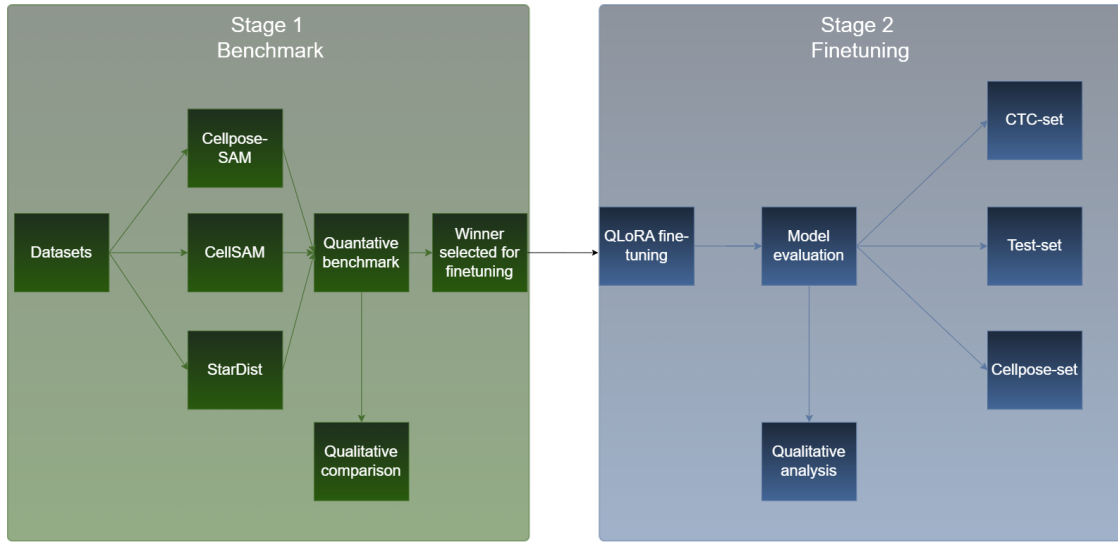


Figure 1.1: Project outline of the benchmarking and fine-tuning stage. The three models, Cellpose-SAM, CellSAM, and StarDist, were evaluated on 18 datasets comprising 4 different modalities. The best performing of these models was then fine-tuned and evaluated against the original model based on three different datasets.

1.4 Limitations

This thesis has imposed several limitations to accommodate the project's scope. Manually inspecting each image in the benchmark datasets would be infeasible. Therefore, partially incorrect masks may be present, which could lead to misleading performance. The benchmark was furthermore restricted to datasets not used in pre-training of CellSAM and Cellpose-SAM, which reduced the number of available evaluation datasets but ensured that reported performance reflects generalisation to unseen data rather than memorisation of training distribution. Furthermore, the benchmarking and fine-tuning did not include histopathology images, as the main focus was on segmenting cells rather than analysing relationships between cells.

2

Preliminaries

This chapter introduces the foundational concepts that underlie both the benchmarking and fine-tuning studies presented in subsequent chapters. First, the four imaging modalities represented in the evaluated datasets are described, providing the visual context necessary to interpret model behaviour across diverse imaging conditions. Next, the Vision Transformer (ViT) architecture is introduced, as it forms the backbone of the FMs evaluated in this thesis. Finally, the evaluation metrics used throughout the project to quantify detection and segmentation performance are defined.

2.1 Imaging Modalities

To capture a wide array of cells in various conditions, different imaging modalities exist to efficiently visualise cellular structures. This section introduces the imaging modalities included in the project, namely fluorescent, brightfield, phase-contrast and differential interference contrast microscopy with examples seen in Figure 2.1.

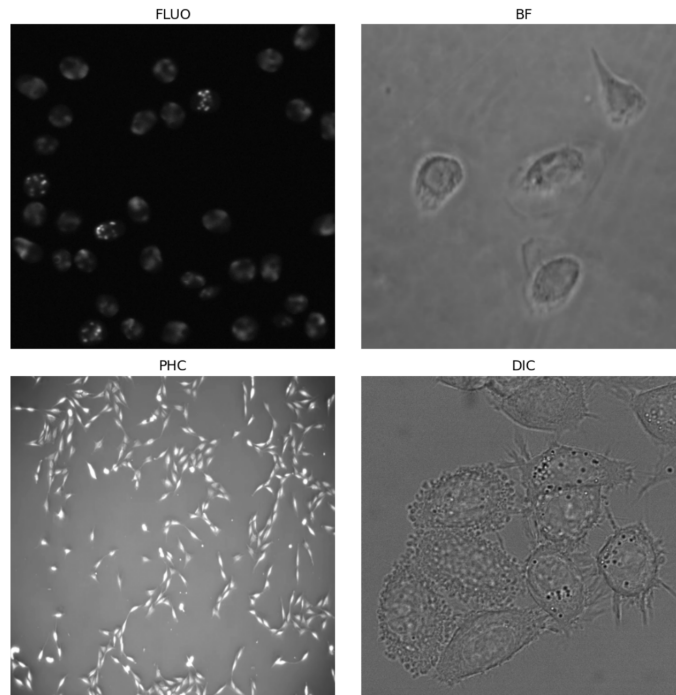


Figure 2.1: Examples of different microscopy modalities: fluorescence microscopy (FLUO-N2DH-SIM+), Phase contrast (PHC-C2DL-PSC), and differential interference contrast (DIC-C2DH-HELA) imaging, all sourced from the Cell Tracking Challenge (CTC) [2]; and brightfield microscopy (BF) from the EVICAN dataset. [13]

Fluorescence microscopy (FLUO) utilises fluorescence to generate images. By illuminating the sample with light at a specific excitation wavelength, molecules absorb the light and, within nanoseconds, emit light at a longer wavelength [14]. Because the excitation and emission wavelengths differ, optical filters can block the excitation light and only transmit the emitted signal. This results in an image where even single fluorescent molecules are detectable if the background has no autofluorescence. Additionally, fluorescent agents, such as dyes or stains, can be added to a sample to target specific structures of interest.

Brightfield microscopy (BF) is a simple method where light is transmitted through the sample and contrast is generated primarily by the increased absorption of light in dense areas [15]. The transmitted light is collected by a lens and focused onto an image sensor, which is then visualised. As many biological samples are transparent or exhibit low contrast, stains are often employed to increase the light absorption of the samples and thus the contrast.

Phase-contrast microscopy (PHC) is useful for observing unstained samples [16]. Compared to BF microscopy, which essentially measures the difference in amplitude of transmitted light, phase-contrast microscopy measures the phase difference, something that is invisible to the human eye. An annular ring, an opaque disk with a transparent circular opening, blocks the centre of the light beam. The light is thereby forced into a hollow cone onto the sample, causing the light to shift in phase. However, this shift is minimal and insufficient to create a clear difference;

therefore, a phase ring is applied. Light that interacts with the sample is diffracted, spreading across the focal plane and largely bypassing the phase ring. Undiffracted background light maintains its path and passes through the phase ring. The phase ring shifts the phase of the background light, and the resulting interference between these two beams converts the phase difference into a visible contrast in intensity.

Differential interference contrast microscopy (DIC) is another technique used without staining. DIC utilises a polariser which polarises the light such that it can be split into two orthogonally polarised beams by a prism, as seen in Figure 2.2 [15]. These beams are separated by a small distance, known as the shear distance, and pass through adjacent regions of the specimen. As they pass through regions with different local refractive indices, each beam experiences a different optical path length difference. A second prism, placed behind the objective, recombines the beams, and another polariser (the analyser) interferes them. The resulting intensity variation then corresponds to the gradient of the optical path length across the specimen.

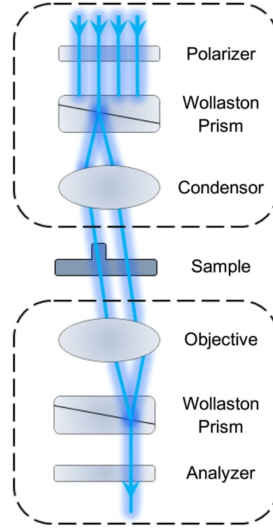


Figure 2.2: Overview of DIC microscopy. Polarised light enters a prism, which splits it into two separated, orthogonally polarised beams. The beams traverse slightly different optical paths through the sample. A second prism recombines them, and the analyser causes interference. The resulting intensity variation reflects the optical path length gradient across the sample. Modified from Heydarian et al. [17] under a CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

2.2 Vision transformers

As the FMs evaluated in this thesis utilise vision transformers (ViT) as their backbones, a short description of their fundamentals is introduced in this section. The ViT adapts the standard Transformer architecture to image data by treating an image as a sequence of fixed-size patches [18]. An input image $x \in \mathbb{R}^{H \times W \times C}$ is divided into $N = HW/P^2$ non-overlapping patches $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$, where C is the

number of channels, H, W the image height and width, and (P, P) is the resolution of each patch. Each patch is flattened into a single vector and mapped to a constant latent dimension D via a trainable linear projection. Because all patch vectors are fed into the transformer simultaneously without an inherent sense of sequence, the 2D spatial order of the image is lost. To combat this loss of spatial information, a learnable position embedding $E_{\text{pos}} \in \mathbb{R}^{(N+1) \times D}$ is added to the patch embeddings, encoding the position of a patch in the image. Additionally, a learnable classification token (x_{class}) is prepended to the sequence, serving as a global representation vector that aggregates information from all patches via self-attention and is used for downstream tasks such as classification:

$$z_0 = [x_{\text{class}}; x_p^1 E; \dots; x_p^N E] + E_{\text{pos}}. \quad (2.1)$$

Note however that not all modern ViT variants employ a classification token. Architectures like masked autoencoders (MAE) can omit this token entirely, opting for global average pooling to aggregate global context [19].

The resulting sequence z_0 is passed through L Transformer encoder blocks. Each block applies multi-head self-attention (MSA) followed by a two-layer multilayer perceptron (MLP), both preceded by layer normalisation and connected via residual connections [18]. In MSA, the input sequence is linearly projected into queries, keys and values across h parallel attention heads, where each head computes scaled dot-product attention, and the outputs are concatenated and projected back to dimension D [20]. This allows the model to jointly attend to information from different representation subspaces at different positions. While the ViT was originally proposed for image classification, subsequent work showed its effectiveness as an encoder backbone for prediction tasks such as segmentation [21].

2.3 Evaluation Metrics

To assess model performance quantitatively, several standard metrics are utilised throughout this project. For segmentation tasks, three metrics evaluating object detection are used. In these metrics, a prediction is counted as a true positive (TP) if its overlap with the ground-truth mask exceeds a specified intersection over union (IoU) threshold; otherwise it is counted as a false positive (FP), and any unmatched ground-truth instance as a false negative (FN).

The primary detection metric is average precision (AP), termed as such in the context of Cellpose-SAM. Mathematically equivalent to an instance-level Jaccard Index, it requires each prediction to meet a specific pixel-wise IoU threshold to be counted as a true positive, and is defined as [7]:

$$\text{AP} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}. \quad (2.2)$$

To evaluate segmentation quality on detected cells independently of detection accuracy, a centroid-matching approach is employed. Ground-truth and predicted cell centroids are extracted, and a distance threshold estimated from the mean cell radius

in the image is used to determine valid matches. Matched pairs are then assigned optimally using the Hungarian algorithm. For each matched pair, the pixel-wise IoU is computed as:

$$\text{IoU} = \frac{|M_{\text{GT}} \cap M_{\text{pred}}|}{|M_{\text{GT}} \cup M_{\text{pred}}|}, \quad (2.3)$$

where M_{GT} represents the ground truth mask and M_{pred} the predicted mask. To calculate the segmentation accuracy across the image, these individual IoU scores are aggregated into an instance IoU (iIoU). The iIoU is defined as the mean IoU across all matched pairs.

Two additional detection metrics, F_1 -error and Error rate, are included to facilitate comparison with prior work. The F_1 -error, used in CellSAM [6], is defined as:

$$1 - F_1 = 1 - \frac{2 \text{TP}}{2 \text{TP} + \text{FP} + \text{FN}}, \quad (2.4)$$

and compared to AP, the F_1 -score weights true positives more aggressively by counting them twice in the denominator. Finally, Cellpose-SAM reports an Error rate, defined as:

$$\text{Error rate} = \frac{\text{FN} + \text{FP}}{\text{TP} + \text{FN}}, \quad (2.5)$$

which measures the total number of detection errors relative to the number of ground-truth instances.

3

Benchmarking

Building on the foundations established in Chapter 2, this chapter presents a systematic evaluation of state-of-the-art segmentation models across diverse microscopy datasets. The three models under evaluation, CellSAM, Cellpose-SAM, and StarDist, are first introduced. The benchmarking methodology is then described, covering dataset selection and the evaluation procedure. Finally, the results are presented and discussed, including a failure case analysis that identifies the conditions under which each model degrades.

3.1 Segmentation models

Recent advancements in FMs have produced a range of specialised adaptations for tasks such as cell-segmentation and tracking [6][7]. This section introduces state-of-the-art (SOTA) models in object segmentation and tracking.

3.1.1 SAM

The Segment Anything Model (SAM) from Meta is an FM trained on 11 million diverse and high-resolution natural images [22]. SAM is built on three different components: an image encoder, a prompt encoder, and a mask decoder. The image encoder is based on a pre-trained ViT that extracts features from the image. The prompt encoder takes as input prompts such as texts or boxes in the image. The mask decoder then utilises the image and prompt embeddings to output a mask. This enables SAM to segment, as the name suggests, a wide variety of images. However, in domains with limited labeled data, such as microscopy, performance may be substantially reduced.

3.1.2 CellSAM

CellSAM is a specialised adaptation of SAM, designed to automate cellular segmentation by replacing manual prompting with an integrated detection module called CellFinder [6]. The architecture, seen in Figure 3.1, utilises a pipeline where images are first passed through a transformer encoder to generate image features. These features are then sent to two different modules. The first module, CellFinder, is a modified Anchor DETR [23] object detector where the backbone is replaced with the fine-tuned SAM ViT. AnchorDETR treats object detection as a set prediction

task by utilising localised 2D anchor points to construct its transformer decoder queries. This set prediction design allows the model to predict multiple, potentially overlapping bounding boxes simultaneously. This capability is highly beneficial in microscopy images, where cells frequently overlap and exhibit high density. Finally, CellFinder converts the processed image features into coordinate bounding boxes, which are passed as bounding box prompts into SAM’s mask decoder.

The second module combines the image features with the bounding box prompts from CellFinder to create masks using SAM’s mask decoder [6]. To ensure compatibility between the detection backbone and SAM’s mask decoder, the neck of the SAM model was fine-tuned. This neck aligns the high-dimensional ViT features with the SAM prompt encoder and mask decoder. During inference, the bounding boxes generated by CellFinder serve as automated prompts, allowing the model to generate individual segmentation masks for thousands of cells simultaneously without human intervention.

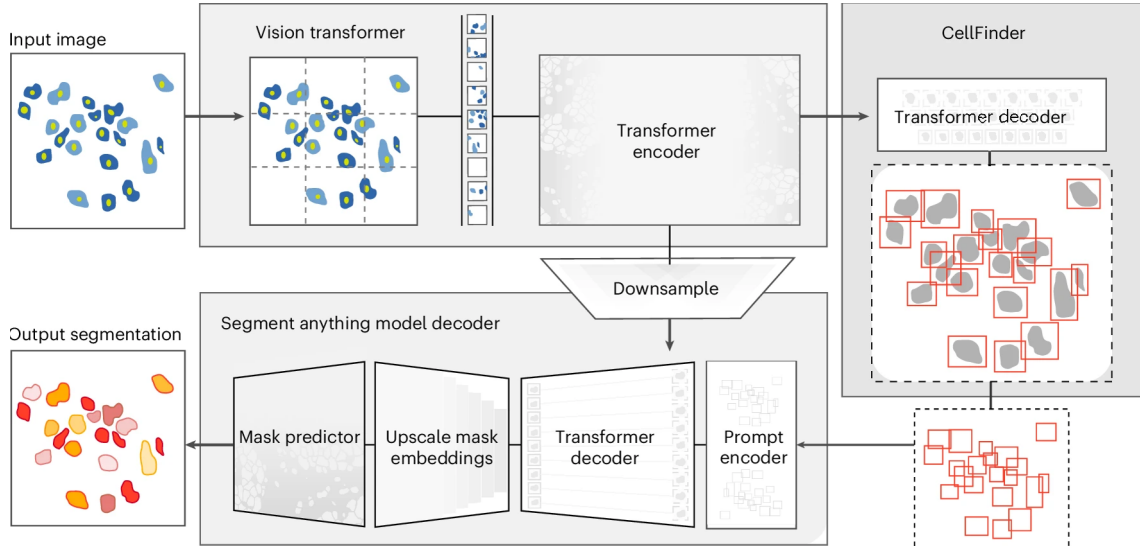


Figure 3.1: Overview of the CellSAM architecture where an input image is sent through a fine-tuned SAM ViT backbone. The features are then sent into both CellFinder, a modified AnchorDETR that detects and creates bounding boxes of cells, which are then sent into the SAM decoder as prompts together with the backbone features. The decoder then generates segmentation masks of cells found in the prompted bounding boxes. Image reprinted from M Marks et al. [6] under a CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

3.1.3 Cellpose-SAM

Cellpose-SAM integrates a modified version of the SAM ViT-Large encoder with the specialised vector flow representation of the original Cellpose algorithm [7]. This architecture differs from the original Cellpose, which used a U-Net. The model utilises the transformer-based encoder to extract image features, which are then mapped to three output channels: horizontal flow, vertical flow, and cell probability, seen in Figure 3.2.

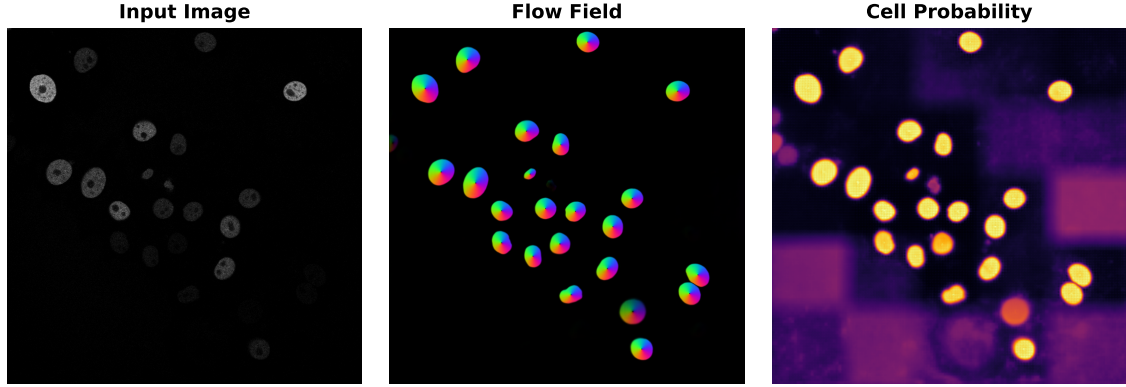


Figure 3.2: Illustration of the Cellpose-SAM segmentation. From left to right: the input image, the predicted flow field, and the cell probability map (high probability shown in yellow, low probability in dark blue/purple)

Compared to other models trained to predict semantic masks, Cellpose-SAM is trained to predict a so-called gradient vector field. A ground-truth spatial gradient representation is derived from annotated cell masks via heat diffusion simulation [4]. In this process, *heat sources* are placed at the centers of each cell, and the heat dissipates through iterations until equilibrium is reached. The gradients of this resulting energy map are then used as ground-truth. The model is then trained to predict both the spatial gradients and per-pixel cell membership. During inference, pixels are treated as particles that *flow* along the predicted gradients toward the respective cell’s center [4]. Because each cell has its own isolated convergence point, touching or overlapping cells can be mathematically separated, as their internal flow patterns point in different directions.

3.1.4 StarDist

StarDist, compared to Cellpose-SAM and CellSAM, is a smaller model built on a U-Net backbone [24]. On top of this backbone, two heads are added. The first head predicts the object probability for each pixel, which is defined as the normalised Euclidean distance to the nearest background pixel where a higher distance yields a higher object probability. This encourages non-maximum-suppression (NMS) to favour pixels near the centre of an object as they generally have a longer distance to the boundary. The second head regresses the radial distances from each pixel to the object’s boundary along a set of n predefined, equidistant angles. During inference, pixels with a high object probability generate polygon proposals based on these radial distances. NMS is then applied to these proposals to resolve redundant detections, resulting in the final instance segmentation.

3.2 Method

This study evaluates two FMs for automated microscopy analysis: CellSAM and Cellpose-SAM, alongside StarDist as a traditional baseline. The methodology proceeds in two steps: benchmarking both models on unseen datasets spanning diverse

cell types and imaging modalities, and then analysing failure cases and why they appear. Performance is quantified using detection metrics (Average Precision (AP) (2.2), F_1 -error (2.4), and Error Rate (2.5)) and centroid-matched instance IoU (iIoU) for pixel-wise segmentation quality (2.3).

3.2.1 Datasets

To compare the performance of the models, 18 different datasets containing previously unseen data were used: a collection of datasets from the CTC [2], **Dynamic Nuclear Net** [25], **CellbinDB** [26], **Aitslab** [27], **TYC** [28], **Revvity** [29], **HepG2** [30], **EVICAN** [13], and **Yeast in brightfield** [31]. The datasets contain images of animal, human, and yeast cells, with fluorescent, brightfield, DIC, and phase-contrast modalities. Each dataset, along with the number of images used, is shown in Figure 3.3.

The masks of the datasets were analysed to determine quality and remove substantially erroneous masks. However, manual inspection of all masks was not feasible. Instead, the metrics AP and iIoU were used to identify potentially incomplete masks by manually analyzing cases where both models failed. As a result, cases containing partially missing masks may be present in the datasets. This was deemed acceptable, as both models would be equally penalised, but overall performance may appear lower.

As both models are trained using their own pre-processing pipeline, no additional pre-processing was utilised except on the **Yeast in brightfield** and **Aitslab** datasets. As the **Yeast in brightfield** dataset contained binary masks, watershed was used to generate an instance segmentation mask from the binary mask. The **Aitslab** dataset, on the other hand, contained faulty ground truth masks, which were corrected using watershed to produce usable instance labels. The scikit watershed function was used, where watershed is an algorithm that is able to identify and separate objects in an image [32].

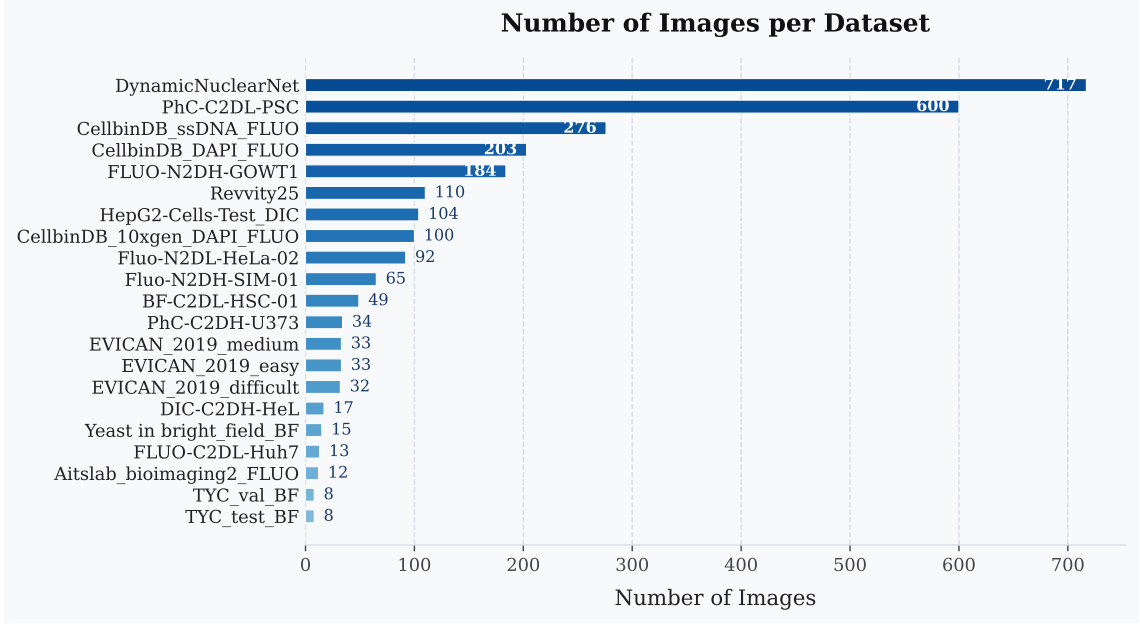


Figure 3.3: Number of images from each dataset used in the benchmarking.

3.2.2 Benchmark procedure

The models were compared with their original evaluation pipeline, with a minimal adjustment for CellSAM, where their percentile normalization was adjusted to utilise the 1st and 99th percentiles. The metrics average precision (AP) (2.2), F1-error (2.4), error rate (2.5), and instance IoU (iIoU) (2.3) were used to compare performance. AP, error rate, and F1-error were calculated for object detection, whereas iIoU was calculated pixel-wise with centroid detection. AP was measured with the thresholds 0.5, 0.75, and 0.9, while the F1-error and the error rate were only measured at threshold 0.5.

As CellSAM is not trained on DIC images, a modality-specific comparison was conducted to avoid bias towards Cellpose-SAM. Through this comparison, initial failure cases revealed that some datasets containing images with uneven lighting or low contrast significantly affected the performance. To evaluate these problems, two contrast-enhancing techniques, specifically Contrast Limited Adaptive Histogram Equalization (CLAHE), and Gaussian blurring were utilised. While CLAHE is integrated into the CellSAM pipeline, it was manually added to the Cellpose-SAM workflow. Finally, an investigation of the impact of the diameter parameter in Cellpose-SAM was conducted to assess its handling of large, out-of-distribution cells, which the model occasionally failed to detect in high-resolution images.

StarDist was included as a traditional baseline to contextualise the performance of the two FMs. Two pre-trained configurations were evaluated: *2D_versatile_flow* for single-channel 2D datasets, trained on the Data Science Bowl 2018 dataset, and *2D_versatile_he* for RGB images, trained on the MoNuSeg 2018 and TNBC dataset [24]). Both configurations were run using the default inference settings without additional preprocessing, since StarDist incorporates its own internal normalization.

3.3 Results

The performance of the models across the benchmarking datasets is presented in this section. Furthermore, the result and cause of the identified failure cases are shown.

3.3.1 Model comparison

The benchmarking demonstrates that FMs adapted for microscopy segmentation generally outperform the traditional baseline StarDist in zero-shot settings. Among the FMs, Cellpose-SAM consistently outperforms CellSAM across the majority of datasets and metrics. The per-dataset quantitative results are summarised in Table B.2, with aggregate performance across all datasets presented in Table B.1. Performance broken down by imaging modality is shown in Figure 3.4 and Figure 3.5.

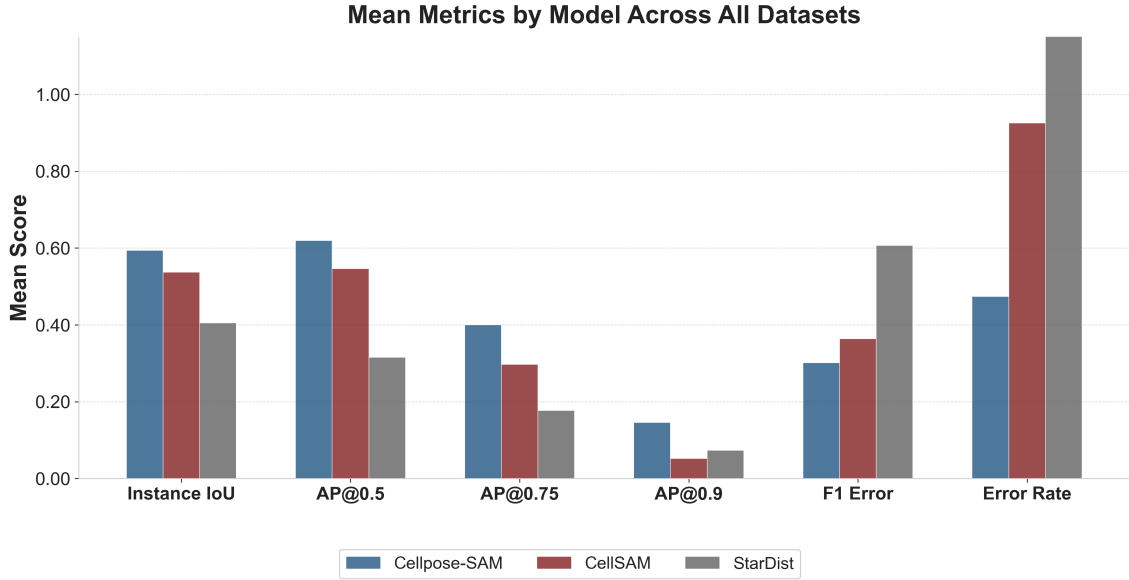


Figure 3.4: Comparison of mean segmentation performance across all benchmarking datasets. Cellpose-SAM outperforms both CellSAM and StarDist across all metrics.

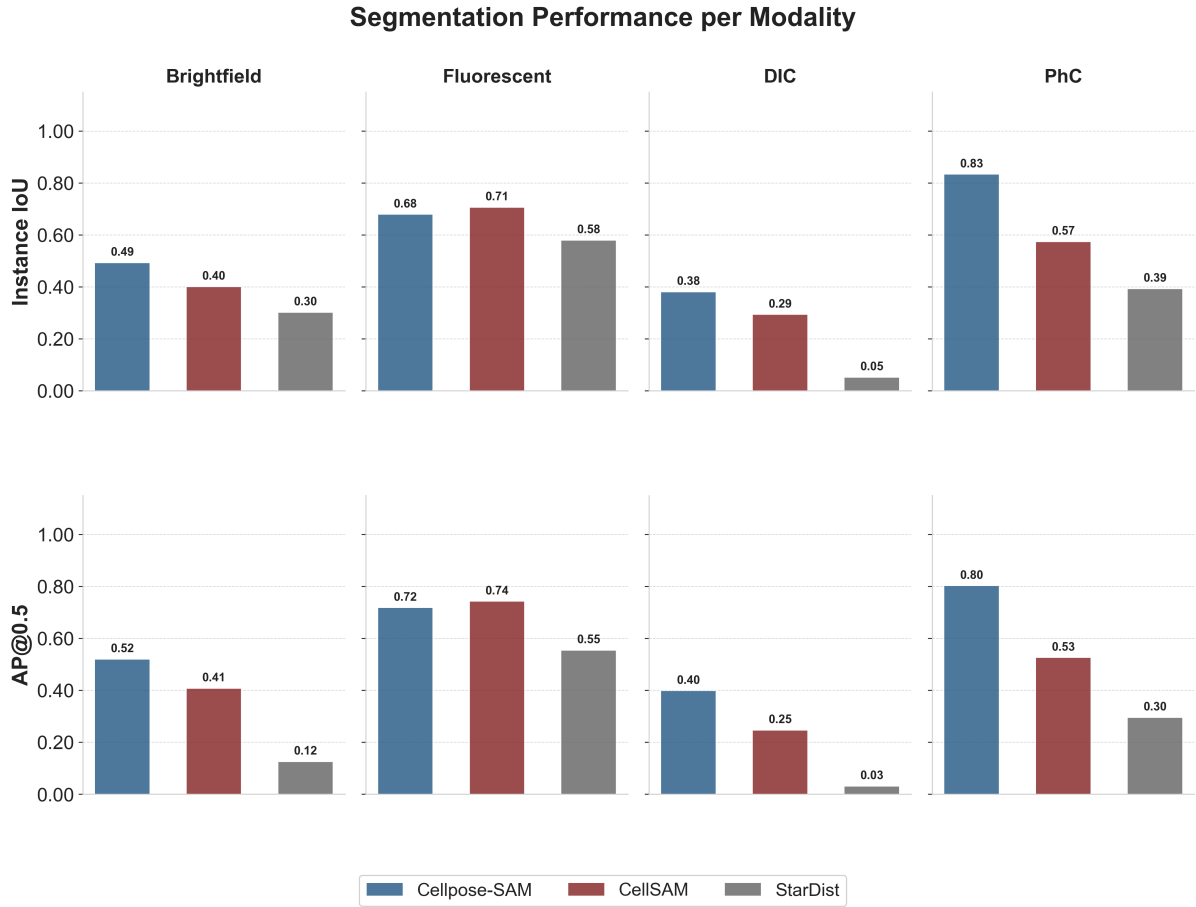


Figure 3.5: Mean instance IoU and AP@0.5 across all benchmarking datasets, split per imaging modality. Cellpose-SAM outperforms both CellSAM and StarDist with the exception on fluorescence images where CellSAM exhibits the best performance by a small margin.

Figure 3.6 illustrates the general performance trade-offs between Cellpose-SAM and CellSAM. While Cellpose-SAM yields more precise object boundaries, it is prone to false negatives, as seen by the green unsegmented cell in the overlay. Conversely, CellSAM exhibits higher sensitivity, missing fewer cells, but tends to over-segment and predict false positives. This is seen as red contours around the cells in the overlay and the erroneous segmentation of background noise as cellular instances.

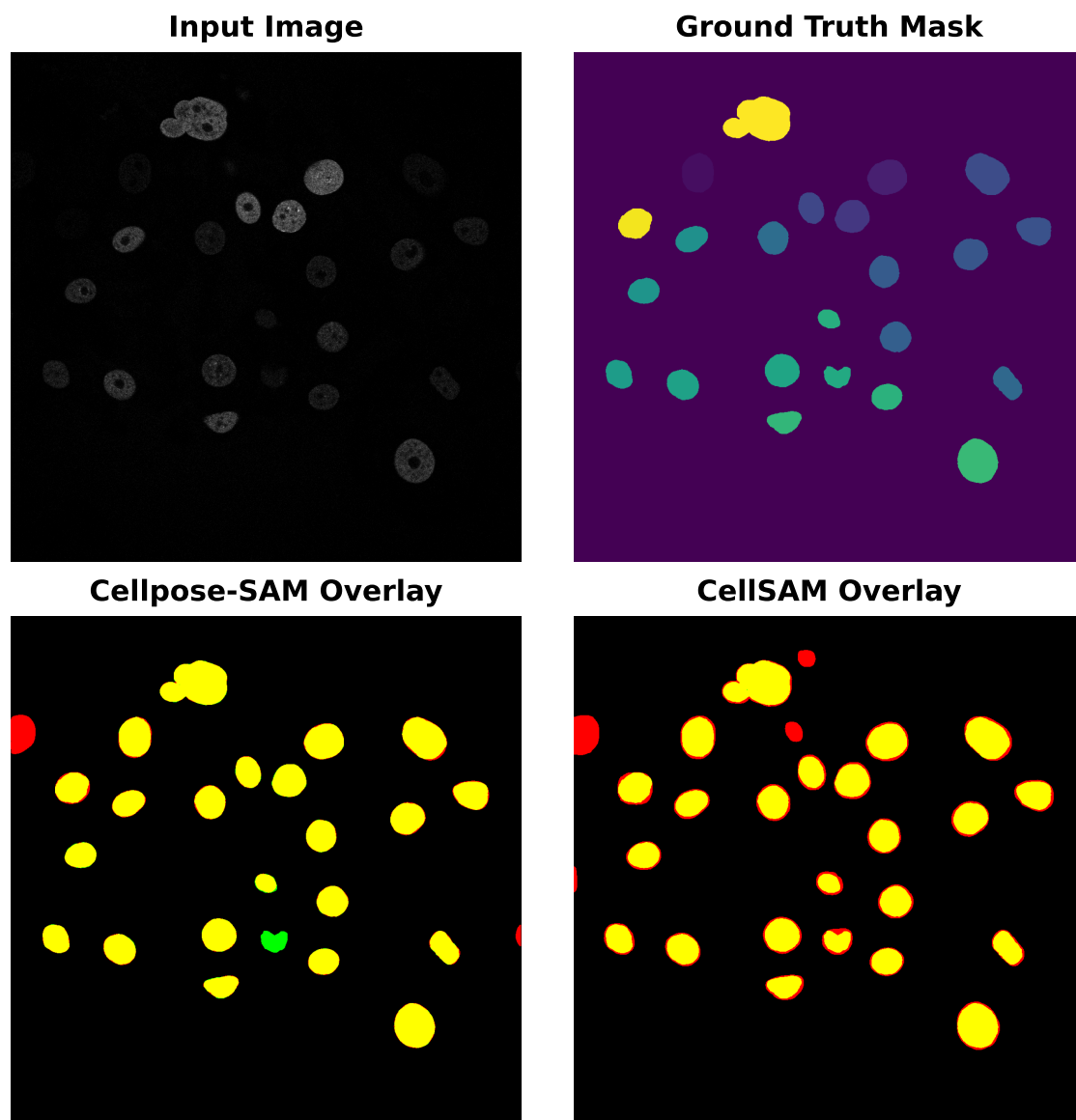


Figure 3.6: Comparison of segmentation accuracy between Cellpose-SAM (left) and CellSAM (right) on an image from the Fluo-N2DH-G0WT1 dataset from the CTC, with FP in red, FN in green, and TP in yellow. Cellpose-SAM shows a superior segmentation accuracy compared to CellSAM, which finds all cells but overpredicts to a greater extent.

3.3.2 Failure cases

In Figure 3.7, the impact of low contrast or uneven lighting on segmentation performance for Cellpose-SAM is shown. Although Cellpose-SAM is trained to handle images with low contrast, there are cases with uneven lighting or low contrast that severely inhibits the model’s segmentation performance. Figure 3.7 illustrates three cases of images with these characteristics and the effect of applying Gaussian blur or CLAHE to increase the contrast.

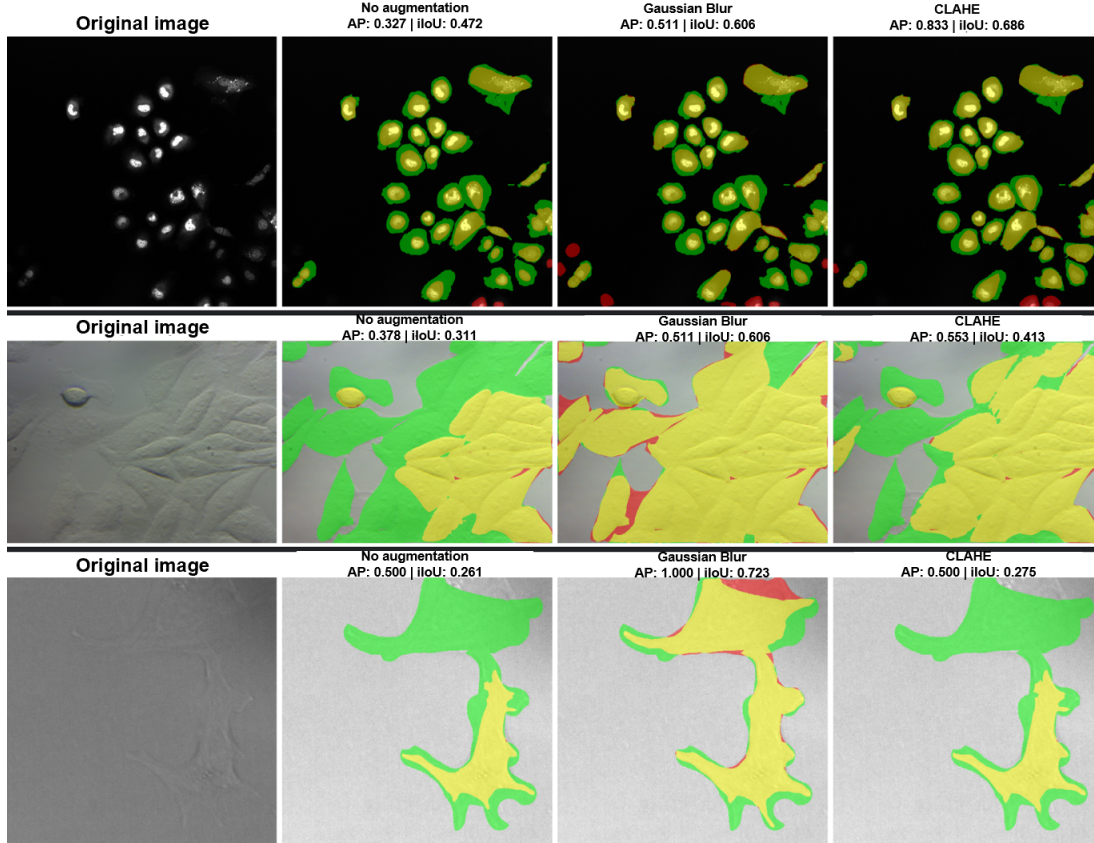


Figure 3.7: Comparison of the segmentation performance of Cellpose-SAM without contrast enhancement, with Gaussian Blur, and with Contrast Limited Adaptive Histogram Equalization (CLAHE). Evaluated on three different images with low contrast and uneven lighting. True positives (TP) are shown in yellow, false positives (FP) in red, and false negatives (FN) in green. Significant improvement can be seen by applying Gaussian Blur on images with uneven lighting and CLAHE on images with low contrast.

Figure 3.8 illustrates the sensitivity of Cellpose-SAM to the diameter parameter on a Revvity BF sample. Without specifying a diameter, the model fails entirely, predicting zero cells. Crucially, the flow and cell probability maps reveal that the internal representations focus on small circular artifacts in the image rather than the actual cells. The cell probability activations remain uniformly low as a result, producing no segmentation output. Providing the mean ground truth cell diameter (139.8 px) recovers the majority of the cell population, yielding an AP@0.5 of 0.870

and an instance IoU of 0.689, with coherent flow gradients and well-defined cell probability activations. This demonstrates that the underlying segmentation capability is intact, but entirely gated by correct scale information – the out-of-distribution diameter of the cells is therefore the primary failure mode on this dataset.

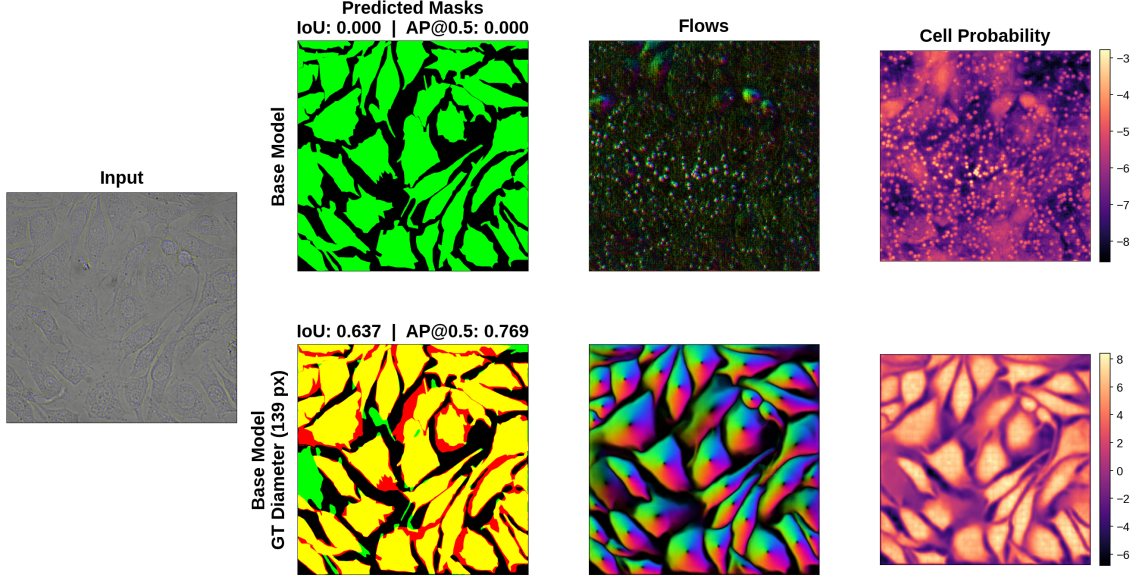


Figure 3.8: Comparison of Cellpose-SAM segmentation performance on an image from the dataset Revvity without manually specifying the diameter parameter (top row), and with a manually set diameter (bottom row). Without setting a diameter, the cells are too large, resulting in the model predicting background noise instead. But by specifying the diameter, the model finds and segments the majority of the cell population.

3.4 Discussion

This section discusses the findings of the benchmark evaluation and the rationale for selecting Cellpose-SAM as the model to fine-tune. Furthermore, the detected failure cases and their solutions are discussed.

As the performance of Cellpose-SAM, seen in Figure 3.4, and Figure 3.5, broadly exceeded that of the CellSAM model, Cellpose-SAM was the selected model to fine-tune. The performance gap is most evident at higher IoU thresholds, reflected in the AP@0.9 metric in Figure 3.4. This superior boundary precision could be a result of Cellpose-SAM’s flow-based representation, where pixels are treated as particles that flow toward cell centers. An example of this is shown in Figure 3.6, where the segmentations from Cellpose-SAM are more precise compared to CellSAM, but with the drawback of being vulnerable to low contrast, a consequence of the Cellpose-SAM pipeline not incorporating any contrast enhancement.

Consequently, contrast enhancing methods such as Gaussian blurring, and CLAHE were introduced. As shown in Figure 3.7, this led to significantly improved performance on datasets containing low contrast images, but reduced performance on

datasets not requiring contrast enhancement. It was therefore not deemed a feasible solution. Furthermore, the solution to manually adjust the diameter parameter on images containing large out-of-distribution cells is not feasible, as the fundamental function of the model is to automatically segment images with no human interference. Thus, fine-tuning was conducted to address these limitations directly.

4

Fine-tuning

This chapter presents the parameter-efficient fine-tuning of Cellpose-SAM, the best-performing model identified in the benchmark. First, the theoretical background for the fine-tuning techniques employed is introduced, covering LoRA, QLoRA, and feature distillation. The methodology, including the dataset construction, model architecture, and training strategy, is then described, followed by the results of the hyperparameter sweep, model selection, and evaluation across held-out, out-of-distribution, and domain retention test sets. Finally, the findings are discussed with respect to fine-tuning strategy, qualitative representation learning, and catastrophic forgetting.

4.1 Fine-tuning techniques

This section introduces the parameter-efficient fine-tuning techniques employed in this work. LoRA and its quantised extension QLoRA form the core adaptation strategy, enabling targeted weight updates with minimal parameter overhead. Feature distillation is additionally explored as a regularisation mechanism to mitigate catastrophic forgetting during fine-tuning.

4.1.1 LoRA

Low-Rank Adaptation (LoRA) is a Parameter-Efficient Fine-Tuning (PEFT) technique designed to optimise the adaptation of large pre-trained models [8]. Instead of updating entire layers or retraining the full model, LoRA introduces two small, low-rank matrices into specific layers of the architecture. During the forward pass, the input is processed by the frozen pre-trained weights in parallel with these adapter matrices:

$$h = W_0x + \Delta Wx, \quad (4.1)$$

here, $W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ represents the frozen weights and x the input vector. The update ΔW is decomposed into two low-rank matrices, $B \in \mathbb{R}^{d_{\text{out}} \times r}$ and $A \in \mathbb{R}^{r \times d_{\text{in}}}$ resulting in:

$$h = W_0x + \frac{\alpha}{r}BAx, \quad (4.2)$$

where the rank r and α are predefined constants and $r \ll \min(d_{in}, d_{out})$, and α is generally set to $2 \times r$. By optimising only A and B via gradient descent, LoRA drastically reduces the trainable parameter count. This targeted optimisation allows the model to learn domain-specific patterns while mitigating catastrophic forgetting of its original knowledge.

4.1.2 QLoRA

Quantised Low-Rank Adaptation (QLoRA) is an extension of LoRA that reduces memory usage enough to fine-tune large models on constrained hardware without sacrificing performance [9]. It achieves this through three key innovations.

The first is a new 4-bit data type called 4-bit NormalFloat (NF4), which is an information-theoretically optimal quantization for normally distributed weights. Since pre-trained neural network weights are typically normally distributed, NF4 quantization preserves more information per bit than standard integer quantization (int4). The second innovation is double quantization, which reduces memory further by quantizing the quantization constants themselves, reducing the average memory footprint of the stored constants. The third is paged optimisers, which use NVIDIA unified memory to handle memory spikes during training by automatically offloading optimiser states to CPU RAM when GPU memory is exhausted.

During the forward and backward pass, the frozen 4-bit weights are temporarily de-quantised to float32 for computation, while the LoRA adapter matrices are kept and trained in float32 throughout. Gradients are thus backpropagated through the quantised frozen model into the adapters, meaning only the low-rank adapter parameters are updated. This separation between the compressed base model and the full-precision adapters is what allows QLoRA to match full fine-tuning performance while dramatically reducing memory requirements. Dettmers et al. demonstrated that a 65B parameter model could be fine-tuned on a single 48GB GPU, a task otherwise requiring multiple high-end GPUs [9].

4.1.3 Feature Distillation

Feature distillation is a technique for preserving learned representations during fine-tuning and preventing catastrophic forgetting [33]. A teacher-student architecture is used, with hooks extracting intermediate features from both models during training. The difference between these features is calculated, in this case, by the mean square error as:

$$L_{distill} = \frac{1}{2} \left(\|f_{block}^S - f_{block}^T\|^2 + \|f_{neck}^S - f_{neck}^T\|^2 \right), \quad (4.3)$$

where (f^S) and (f^T) denote the spatially normalised feature maps for the student and teacher, respectively. In cases where hooks are used to capture features at multiple intermediate locations, an exponential moving average (EMA) can be implemented to act as a dynamic weighting factor, as the features' magnitude may vary at different parts of a model. This ensures that the total distillation loss is not dominated by a single layer. The weighted difference is incorporated into the

training loss, penalising the student for deviating from the teacher’s intermediate representations and thereby discouraging catastrophic forgetting.

4.2 Fine-tuning methodology

Through the comparative benchmark, Cellpose-SAM demonstrated the most promising results and was selected for fine-tuning. This approach combined a curated dataset with parameter-efficient training to ensure an efficient training process.

4.2.1 Dataset

A dedicated tuning dataset, seen in Figure 4.1, was constructed by aggregating images from the previously evaluated datasets. To prevent overfitting toward larger datasets, the number of images sampled from each dataset was capped. An exception was made for the **Dynamic Nuclear Net** [25] and **Revvity** [29] datasets, which contained a large number of difficult cases and were therefore represented more heavily in the tuning dataset. Although the resulting size imbalance between these datasets and smaller ones such as **BF-TYZ_test** was larger than initially intended, no signs of excessive overfitting toward the overrepresented datasets were observed in the fine-tuning results, and the split was therefore retained.

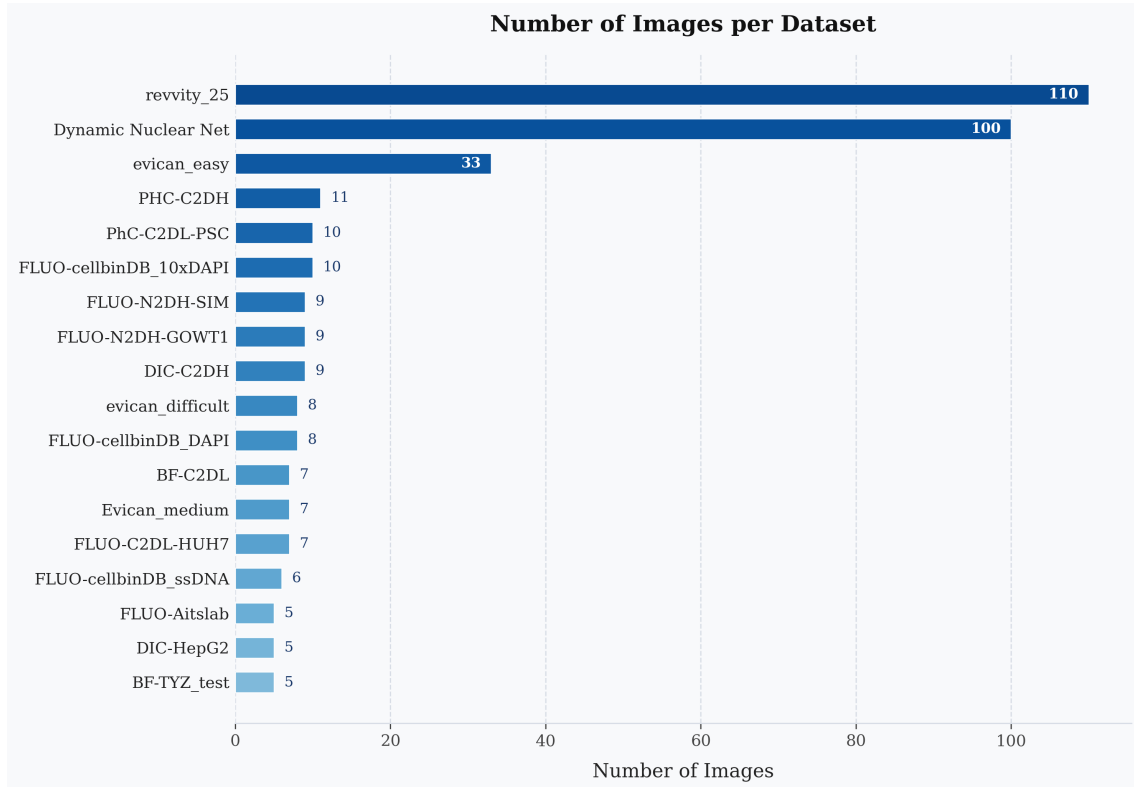


Figure 4.1: Number of images from each dataset included in the curated fine-tuning dataset.

To improve robustness and prevent overfitting, the dataset was categorised into

normal and *hard* cases. Hard cases were found by analysing the worst performing cases in each dataset, measured by the iIoU. Cases where poor performance was attributable to annotation errors rather than model failure were excluded; the remainder were added to the tuning dataset. The definition of "hard" therefore varied between datasets: in some, the worst-performing case had an iIoU of 0.6, while in others it was 0.0. In cases involving video sequences, such as those from the CTC, sequences were strictly separated between training and testing sets to prevent data leakage.

The final dataset was partitioned into a training, validation, and testing set with an initial target split of 80/10/10 (Table 4.1). While random sampling was employed to distribute hard cases evenly, strict constraints forced deviations from this ideal distribution. Specifically, several datasets contained a low image count (some as few as five images), and images originating from the same video sequences had to be kept strictly within the same partition to prevent data leakage. As a result, the final distribution yielded a 73/11/16 split for the training, validation, and test set respectively. These constraints also caused fluctuations in the ratio of hard cases, where the training set had 40.6%, the validation set 38.5%, and the test set 48.3%. While unintentional, this distribution subjects the model to a challenging evaluation phase, ensuring it is robust under difficult conditions.

Table 4.1: Distribution of normal and hard cases across the 80/10/10 dataset split. The discrepancies between the validation and test sets are the result of video sequence separation.

Split	Total	Normal	Hard	Hard (%)
Train	261	155	106	40.6%
Validation	39	24	15	38.5%
Test	58	30	28	48.3%
Total	358	209	149	41.6%

4.2.2 Model architecture and tuning strategies

The fine-tuned Cellpose-SAM architecture utilised a frozen foundation backbone combined with an unfrozen neck, ensuring that the core SAM ViT encoder weights were not strictly updated via standard backpropagation. Initial pilot tests guided two key architectural decisions: first, unfreezing the model neck yielded better performance than a fully frozen configuration. Second, expanding the parameter-efficient fine-tuning to include the entire transformer block, rather than isolating the attention mechanisms, provided a distinct performance boost. Consequently, QLoRA adapters were injected into all linear modules of selected encoder blocks to enable targeted adaptation while preserving pre-trained representations. Specifically, adapters were integrated into the fused query/key/value projection (`attn.qkv`), the attention output projection (`attn.proj`), and both MLP layers (`mlp.lin1`, `mlp.lin2`).

Following the theoretical formulation described in Section 4.1.2, the frozen base weights were quantised to 4-bit NF4. During the forward and backward passes,

these weights were temporarily de-quantised to 16-bit BrainFloat (`bfloat16`) for base layer computations. Meanwhile, the trainable LoRA adapter matrices were maintained and updated in 32-bit precision (`float32`) to ensure numerical stability. This hybrid-precision approach allowed for the targeted adaptation of the large vision encoder on constrained hardware while preserving the pre-trained representations in the frozen backbone.

The feature distillation was implemented by applying hooks to capture the intermediate features from the final transformer block and the neck layer of both the teacher and the student network. To account for potential discrepancies in feature scaling, both sets of features were L_2 -normalised before calculating the distillation loss via the mean squared error, as defined in Equation 4.3. Furthermore, an exponential moving average (EMA) was used to track and adjust the magnitudes of the individual loss components throughout training.

To evaluate generalisability and monitor for catastrophic forgetting, the fine-tuned model was compared against the base Cellpose-SAM on two held-out datasets. The original Cellpose training dataset was used to measure retention of previously learned features, providing a direct signal for forgetting. The CTC Fluo-C2DL-MSD dataset, which was absent from both the fine-tuning process and the original Cellpose-SAM training, served as a measure of out-of-distribution generalisation.

4.2.3 Implementation details and training strategy

Initial fine-tuning experiments were conducted on a single T4 GPU with 16 GB VRAM on Kaggle, and on an NVIDIA RTX4060 with 8GB of VRAM in an in-house setup. The main parameter sweeps were conducted on either an NVIDIA A40 (48 GB VRAM) or A100 (40 GB VRAM) GPU on the Alvis cluster at Chalmers C3SE. The implementation was built using PyTorch and the `bitsandbytes` library for 4-bit quantization. Using the qLoRA configuration, the parameter footprint was reduced, constraining peak VRAM usage and allowing for efficient training.

The network was optimised using AdamW with a weight decay of 0.1. During training, the original Cellpose-SAM training pipeline was used, where input images are dynamically cropped and resized to a resolution of 256×256 pixels. To stabilise early training and ensure optimal convergence, a sequential learning rate scheduler was employed. This consisted of a linear warmup phase for the first 10% of the epochs, increasing the learning rate from a factor of 0.1 to its base value, followed by a cosine annealing schedule that gradually decayed the learning rate to 1% of the initial value by the end of training.

For the feature distillation mechanism, the distillation weight factor was set to 0.2, and the decay rate for the exponential moving average (EMA) dynamic weighting was configured to 0.99. All the models were trained for 30 epochs using PyTorch’s mixed-precision `autocast` with 16-bit BrainFloat (`bfloat16`) to maximise computational efficiency. Total training time was approximately 15 minutes on the Alvis cluster with a batch size of 8. The same configuration could be run on Kaggle and the in-house setup, but required a batch size of 4 to avoid exhausting available

VRAM. The training time for the in-house setup was approximately 2 hours, and for Kaggle, 1.5 hours.

To optimise the selection of hyperparameters, a sequential sweep strategy was employed where each sweep narrows the search space for the next. Initial pilot tests revealed that injecting QLoRA into both the attention and MLP layers, and adapting more than a few encoder blocks, improved performance. Based on these findings, parameters were further optimised through the following sweeps:

Learning Rate: Evaluated the model’s sensitivity to gradient update magnitude to identify a stable training range.

Injection Width: Determine which Transformer blocks should integrate adapters (e.g., blocks 12–23 vs. 0–23), balancing low-level and high-level feature adaptation.

QLoRA parameters and Batch Size: Jointly optimise QLoRA adapter capacity and training stability, as these parameters interact closely.

Distillation Weight: Evaluate the impact of feature distillation losses at the Transformer block and neck outputs (f_{block} , f_{neck}) as a mechanism to mitigate catastrophic forgetting.

Combined Configuration: Evaluate the best-performing setting from each preceding sweep in combination to verify that individual gains are complementary.

While this heuristic approach ensured a computationally efficient search, the high dimensionality of the hyperparameter space meant an exhaustive grid search was not feasible, and global optima outside the explored regions may exist. In particular, the distillation weight and hook placement were not independently optimised, leaving open the possibility that feature distillation may yield further gains with dedicated tuning. As individual model checkpoints were not retained after each sweep, formal statistical significance testing was not performed, and configuration selection was guided by practical considerations rather than marginal metric differences alone.

4.2.4 Fine-tuning evaluation

To robustly compare the fine-tuned configurations, three different datasets were used, aimed at evaluating different characteristics. The held-out test set from the curated fine-tuning dataset (Figure 4.1) measured domain adaptation. The original **Cellpose** test set [4] served as a domain retention control, providing a direct signal for catastrophic forgetting. The **Fluo-C2DL-MS** dataset from the Cell Tracking Challenge, absent from both fine-tuning and the original Cellpose-SAM training, measured out-of-distribution generalisation on fluorescence microscopy. Three additional datasets were evaluated on the final selected configuration to broaden the generalisation assessment across imaging modalities: **ALFI** (DIC) [34], **Phase400** (phase contrast) [6], and **StarDist_BF_Monocytes** (BF) [35]. These were not used during model selection.

4.3 Results

This section presents the fine-tuning results for Cellpose-SAM. The primary finding is that QLoRA adaptation on a curated dataset of 358 images yields substantial improvements on the held-out test set, with all five finalist configurations achieving AP@0.5 between 0.747 and 0.778, compared to 0.539 for the base model. These gains come at minimal cost to general-purpose capability: the fine-tuned model retains near-baseline performance on the original **Cellpose** test set and improves slightly across all metrics on the held-out CTC fluorescence dataset. Supplementary evaluation on **ALFI**, **Phase400**, and **StarDistBF** provides additional generalisation evidence across imaging modalities, though two of the three datasets carry annotation mismatches that complicate direct metric interpretation.

4.3.1 Hyperparameter optimisation

Full per-configuration results for each sweep are provided in Tables C.1–C.4 in the appendix. The findings from each sweep are summarised below.

Learning rate sweep: As shown in Figure C.1, for qLoRA injections in block 12–23, learning rates at or below 1×10^{-5} resulted in underfitting, with AP@0.5 on the test set remaining below 0.72. The highest rate tested (5×10^{-4}) achieved the strongest test set AP@0.5 (0.792) and AP@0.75 (0.613), but performed poorly on the CTC out-of-distribution set (AP@0.5 = 0.150, iIoU = 0.134), indicating overfitting to the training distribution and catastrophic loss of generalisation. A learning rate of 5×10^{-5} offered the best balance, reaching an AP@0.5 of 0.760 and iIoU of 0.710 on the test set while preserving near-baseline performance on the **Cellpose** control set while increasing performance on CTC data.

Block sweep: As shown in Figure C.2 and Table C.2, restricting adapter injection to the final six encoder blocks produced the weakest test set performance (AP@0.5 = 0.701), suggesting that mid-level encoder features contribute to domain adaptation. Injecting adapters across all 24 blocks (0-23) achieved the highest AP@0.9 (0.249) and the best CTC AP@0.5 (0.510), while all block configurations showed only minor regression on the **Cellpose** control set relative to the base model.

Batch size and QLoRA parameters: As shown in Table C.3, reducing the adapter rank from 32 to 16 ($\alpha = 32$) yielded comparable test set AP@0.5 (0.773) and iIoU (0.714) to the higher-rank configuration, while producing better retention on the **Cellpose** control set (iIoU: 0.739 vs. 0.734) and stronger CTC generalisation (iIoU: 0.499 vs. 0.483). This suggests that a lower-rank adapter is sufficient for the target distribution while better preserving the base model’s broader representations. Batch size had limited influence on detection metrics, though a batch size of 1 increased test set performance at the cost of a notable CTC iIoU drop (0.371), suggesting training instability at very small batches.

Combined sweep: The combined sweep combined the lower rank and alpha (rank = 16, $\alpha = 32$) from the Batch/rank sweep with both broader block injection (0-23) and a reduced learning rate (5×10^{-5}) (Table C.4). Rank = 32, $\alpha = 64$ over all blocks at LR 5×10^{-5} achieved the highest test set AP@0.5 in this sweep (0.777, iIoU = 0.711), but at the cost of greater Cellpose regression than the blocks 12-23 configurations. Restricting injection to blocks 12-23 with LR 5×10^{-5} (rank 16, $\alpha = 32$) preserved the strongest generalisation across both control sets (Cellpose iIoU = 0.737, CTC iIoU = 0.515), though with a more modest test set gain (AP@0.5 = 0.747).

Distillation sweep: Feature distillation at the Transformer block and neck outputs was evaluated across configurations mirroring those in the combination sweep (Table C.5). Across the four matched configurations, the effect of distillation on test set AP@0.5 ranged from -0.018 to +0.028, and an instance IoU from -0.007 to +0.051, with no consistent direction of improvement. On the **Cellpose** control set, differences were negligible (≤ 0.002 AP@0.5 across all configs). On the CTC set, distillation configurations showed slightly lower performance in three of four cases, with AP@0.5 differences of up to -0.078. Distillation did not produce consistent gains on any evaluation set.

4.3.2 Model selection

The best-performing configuration from each sweep was taken forward for final comparison. Selection was based primarily on AP@0.5 and instance IoU on the held-out test set, with test set performance weighted over control set performance. Configurations exhibiting substantial degradation on either the **Cellpose** control set or the CTC out-of-distribution data relative to the base model were excluded. The five finalist configurations are compared against the base model in Table 4.2 and Figure 4.2.

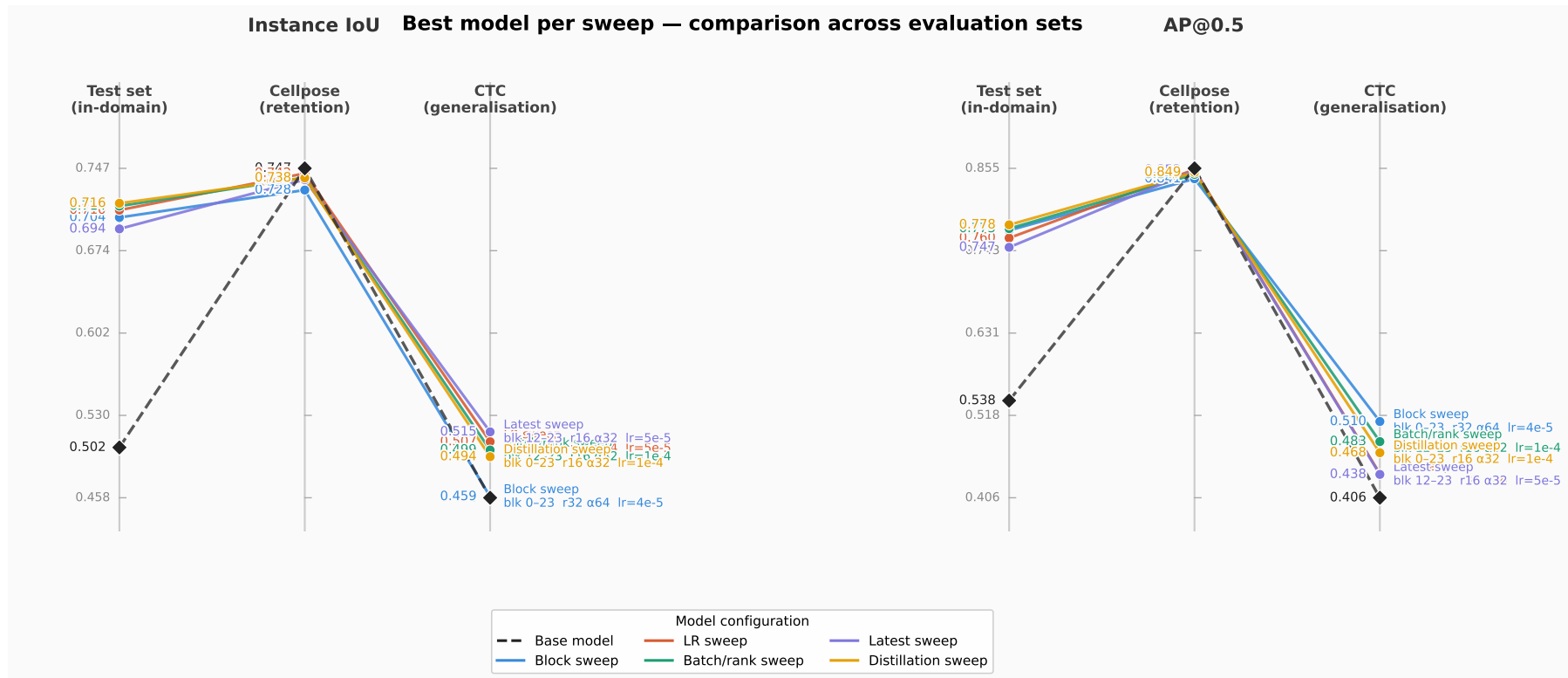


Figure 4.2: Comparison of segmentation performance of the base model and the five best performing configurations on the fine-tuning test set, Cellpose dataset, and Fluo-C2DL-MSC dataset from the CTC, measured by iIoU and AP@0.5.

Table 4.2: Metrics of the base model and the five best performing configurations across the fine-tuning test Set, Cellpose dataset, and the Fluo-C2DL-MSD dataset from the CTC.

Sweep	Blocks	Rank	Alpha	LR	F1 err	Err rate	AP@0.5	AP@0.75	AP@0.9	IoU
TEST SET										
Base model	—	—	—	—	0.3958	0.8559	0.5385	0.3241	0.1088	0.5017
Block sweep	0-23	32	64	4e-5	0.1550	0.5957	0.7703	0.5831	0.2488	0.7037
LR sweep	12-23	32	64	5e-5	0.1611	0.6730	0.7599	0.5596	0.2220	0.7101
Batch/rank	12-23	16	32	1e-4	0.1540	0.6568	0.7727	0.5855	0.2284	0.7136
Latest	12-23	16	32	5e-5	0.1712	0.6934	0.7472	0.5477	0.1936	0.6936
Distillation	0-23	16	32	1e-4	0.1494	0.6210	0.7778	0.5804	0.2359	0.7161
CELLPOSE										
Base model	—	—	—	—	0.0836	0.1602	0.8548	0.6322	0.1669	0.7468
Block sweep	0-23	32	64	4e-5	0.0940	0.1728	0.8406	0.6229	0.1640	0.7277
LR sweep	12-23	32	64	5e-5	0.0851	0.1616	0.8532	0.6368	0.1660	0.7424
Batch/rank	12-23	16	32	1e-4	0.0891	0.1681	0.8465	0.6357	0.1751	0.7387
Latest	12-23	16	32	5e-5	0.0860	0.1619	0.8521	0.6324	0.1515	0.7368
Distillation	0-23	16	32	1e-4	0.0877	0.1635	0.8493	0.6385	0.1680	0.7384
CTC										
Base model	—	—	—	—	0.4596	1.3487	0.4062	0.1438	0.0088	0.4575
Block sweep	0-23	32	64	4e-5	0.3813	1.0839	0.5100	0.1852	0.0150	0.4585
LR sweep	12-23	32	64	5e-5	0.4208	1.3253	0.4390	0.1539	0.0206	0.5067
Batch/rank	12-23	16	32	1e-4	0.3810	1.1734	0.4827	0.1666	0.0178	0.4994
Latest	12-23	16	32	5e-5	0.4242	1.3384	0.4380	0.1597	0.0201	0.5154
Distillation	0-23	16	32	1e-4	0.4108	1.3181	0.4675	0.1645	0.0139	0.4936

All five fine-tuned configurations improve substantially over the base model on the test set: AP@0.5 rises from 0.539 to at least 0.747 and iIoU from 0.502 to at least 0.694, with only modest losses on the **Cellpose** control set. Metric differences between finalist configurations are generally small in magnitude.

The Distillation configuration (Blocks 0-23, rank = 16, $\alpha = 32$, LR 1×10^{-4}) achieved the strongest overall test set performance with an AP@0.5 (0.778), F1 error (0.149), and instance IoU (0.716), while maintaining competitive Cellpose retention. However, the distillation sweep was not conducted as an independent hyperparameter search, since configurations mirrored those from the final combination sweep, meaning the distillation weight λ and hook placement were fixed and not optimised. It is therefore likely that the distillation configuration has not been fully exploited.

The Batch/rank configuration (blocks 12-23, rank = 16, $\alpha = 32$, LR 1×10^{-4}) was selected as the final model, as it matches the distillation configuration’s performance while training only 1.3% of parameters (4,048,064 out of 307,774,658) and avoiding the memory overhead of a teacher forward pass. The performance of the final model against the base model across all three evaluation sets is summarised in Figure 4.3. The distillation configuration is retained as a candidate for future work with dedicated hyperparameter optimisation.

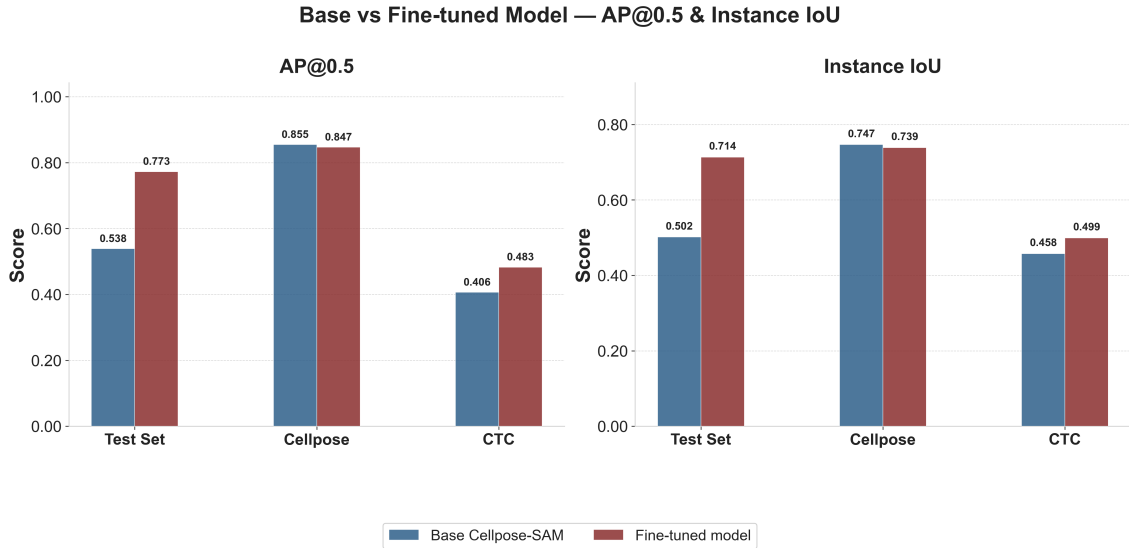


Figure 4.3: Comparison of segmentation performance measured by AP@0.5 and instance IoU between the base Cellpose-SAM and the final fine-tuned model across the fine-tuning test set, **Cellpose** dataset, and the **Fluo-C2DL-MS** from the CTC.

4.3.3 Domain adaptation (Held-out test set)

To evaluate the final model’s domain adaptation, the performance on the held-out test set from the fine-tuning dataset was used. As shown in Table 4.3 and Figure 4.4, performance on the held-out test set improved significantly across all metrics. For the final model, the F1-error is reduced by more than 60% (0.396 to 0.154), and the error rate decreases substantially (0.856 to 0.657). Among the detection metrics,

AP@0.5 and AP@0.75 increase considerably (0.539 to 0.773 and 0.324 to 0.586, respectively), while AP@0.9 more than doubles (0.109 to 0.228). The instance IoU, reflecting segmentation quality on correctly detected cells, increases strongly from 0.502 to 0.714.

Table 4.3: Segmentation performance on the fine-tuning test set. Best result per metric in **bold**.

Model	F1 Error	Error Rate	AP@0.5	AP@0.75	AP@0.9	Inst. IoU
Base Cellpose-SAM	0.3958	0.8559	0.5385	0.3241	0.1088	0.5017
Fine-tuned QLoRA	0.1540	0.6568	0.7727	0.5855	0.2284	0.7136

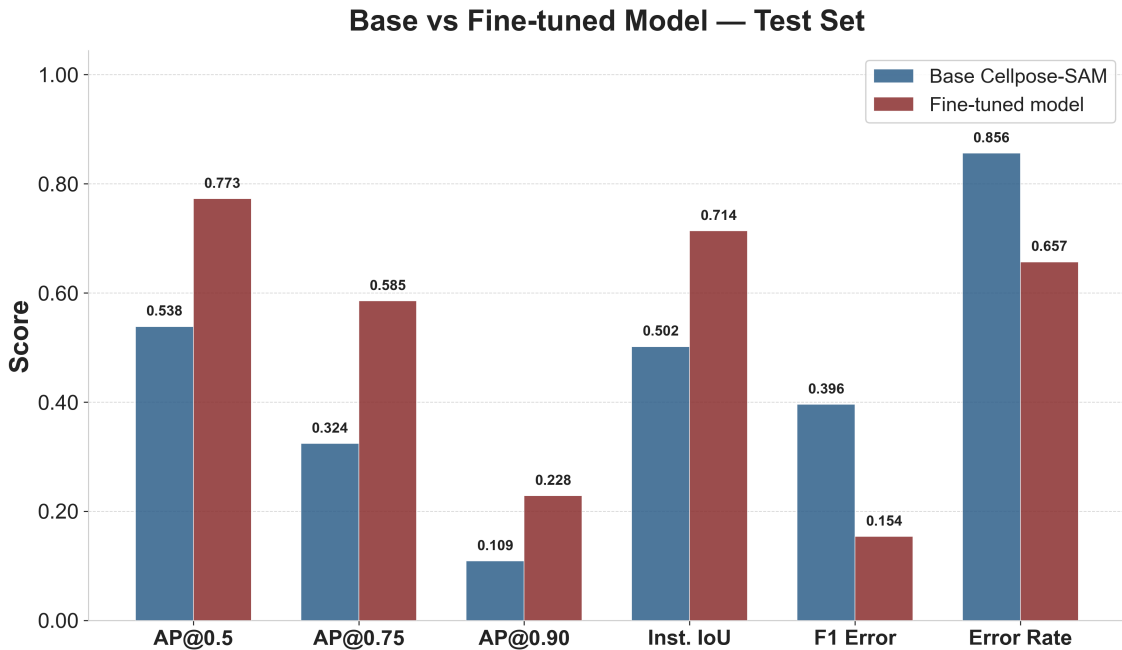


Figure 4.4: Segmentation performance of the base and fine-tuned Cellpose-SAM model on the fine-tuning test set across all six metrics. Shaded columns indicate lower-is-better metrics.

However, the original masks were used during fine-tuning, contributing potentially noisy supervision for the 3 training images from this dataset.

To understand where the fine-tuned model gains and loses performance relative to the base model, Figure 4.5 shows per-dataset Δ IoU. The largest gains are concentrated in the **Revvity** and **DIC-HepG2** datasets, both of which contain low-contrast BF and DIC images that the base model generally failed to segment. It should be noted that watershed was not applied to the **Aitslab** samples during fine-tuning, resulting in incorrect masks and therefore the empty results both for the base and fine-tuned model in Figure 4.5. Figures 4.6-4.10 illustrate representative cases across the test set, showing both losses and improvements. Figure 4.6 shows the most improved sample, a **Revvity** BF image containing large out-of-distribution cells, for

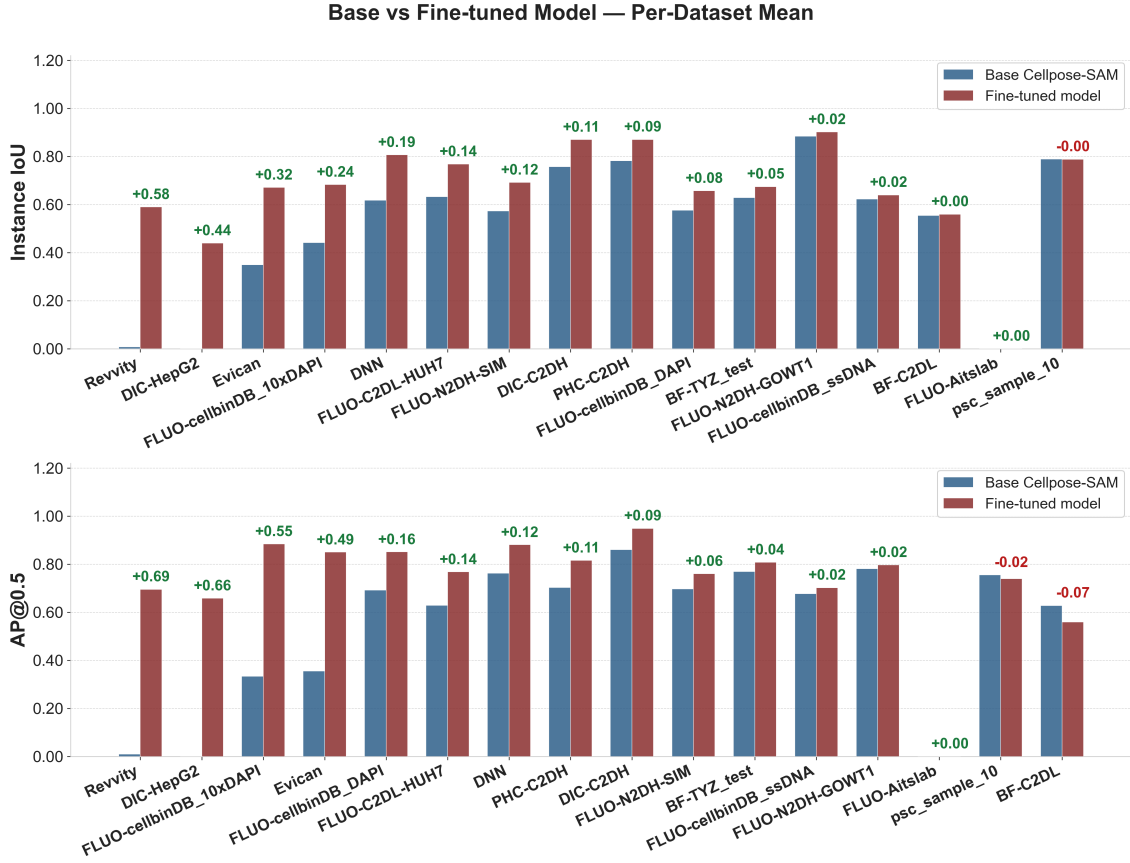


Figure 4.5: Comparison of segmentation performance between the Fine-tuned model and the base model, per dataset in the fine-tuning test set. Sorted by the highest performance difference from left to right. The upper graph shows the difference in instance IoU performance, and the lower, AP@0.5.

the base model, which predicts zero cells, while the fine-tuned model recovers the majority of the cell population. Figure 4.7 shows a PhC-C2DH case where the fine-tuned model is more aggressive with its cell probability threshold, improving instance IoU but introducing false positives that reduce AP@0.5. The top 5 gains and losses across the full test set are shown in Figure E.1 and Figure E.2 respectively.

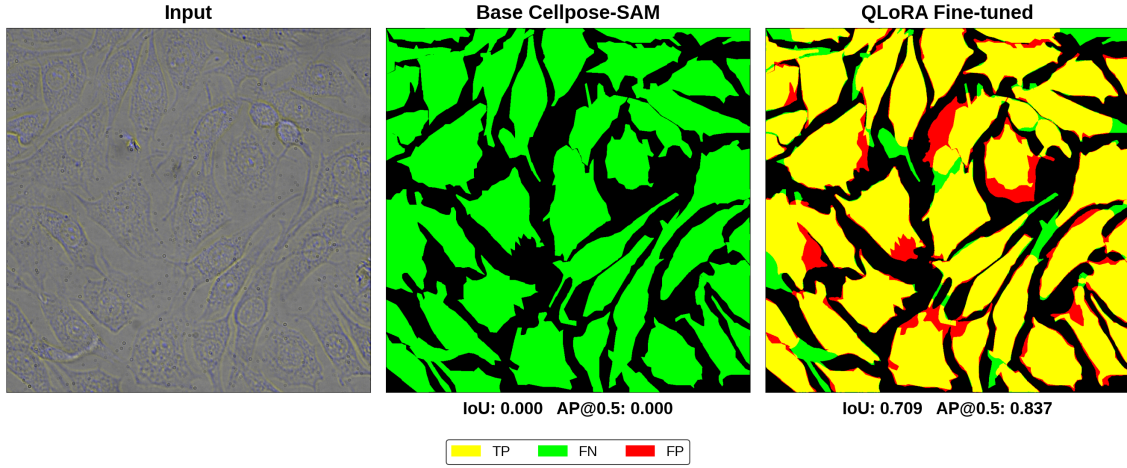


Figure 4.6: Performance of the base- (center) and fine-tuned (right) model on the Revvity image containing cells with an out-of-distribution size. Following fine-tuning, the model, compared to the base model, can find and segment cells where true positives are shown in yellow, false positives in red, and false negatives in green.

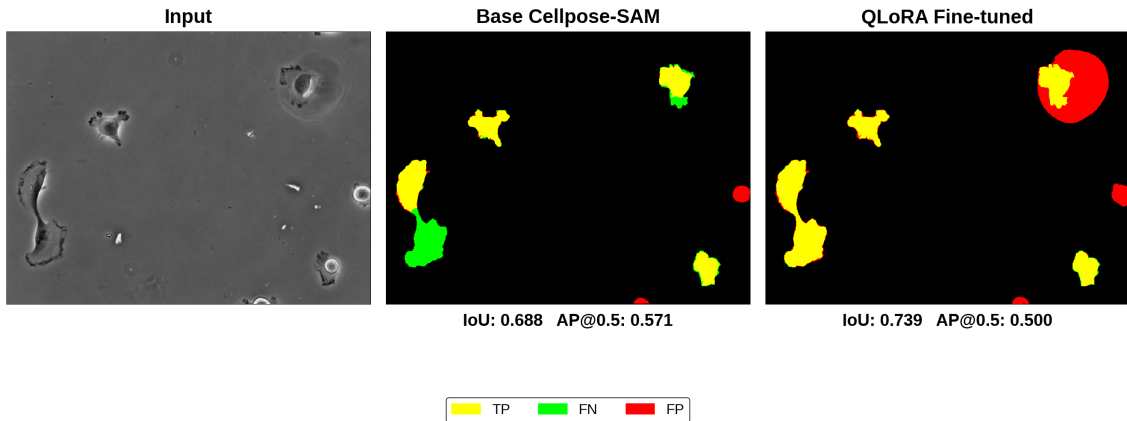


Figure 4.7: Input image, base- and fine-tuned-model segmentation on an image from the dataset PhC-C2DH with true positives in yellow, false positives in red, and false negatives in green. It can be seen that the fine-tuned model finds all cells but also oversegments in its prediction.

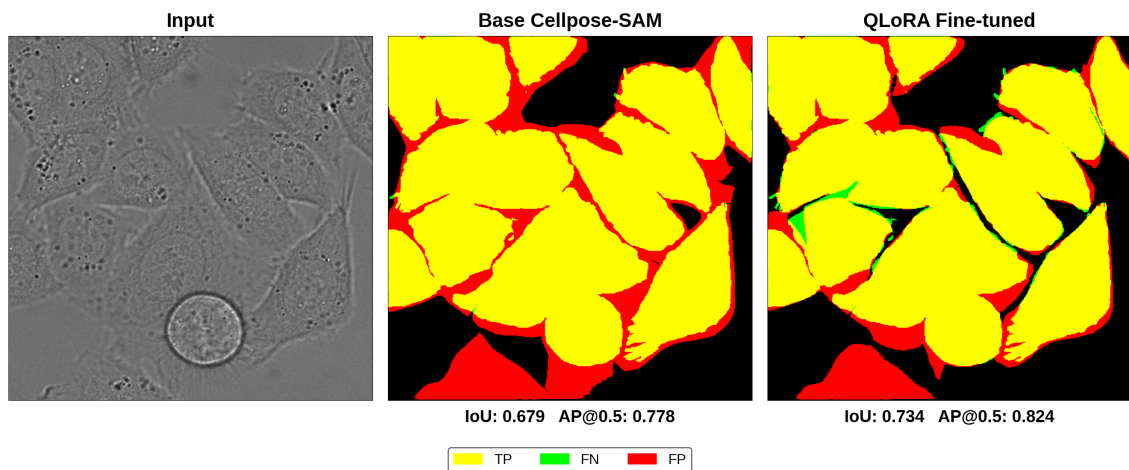


Figure 4.8: Input image, base- and fine-tuned-model segmentation on a DIC-C2DH image with true positives in yellow, false positives in red, and false negatives in green. The fine-tuned model can, compared to the base model, more accurately distinguish between the background and foreground, resulting in more precise segmentations.

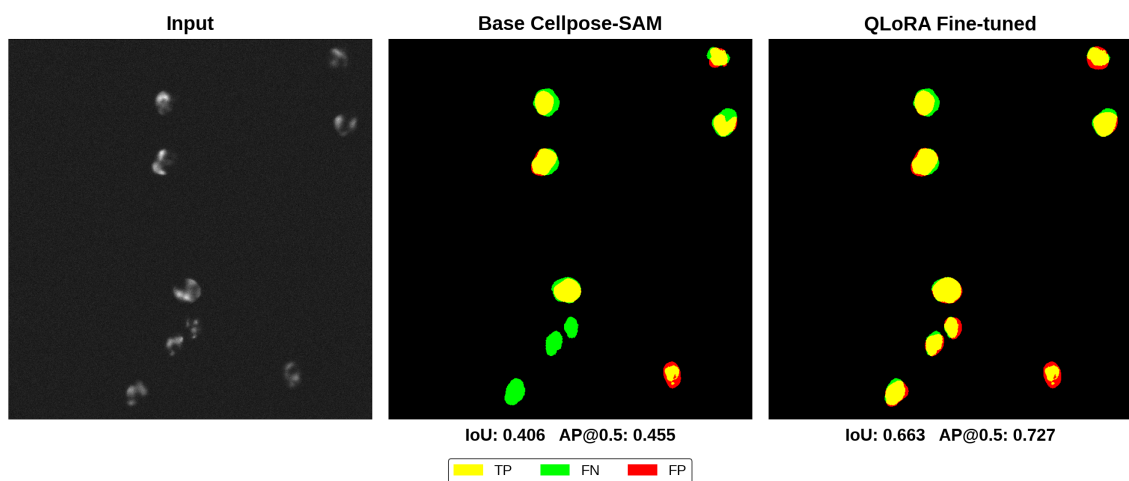


Figure 4.9: Input image, base- and fine-tuned-model segmentation on a Fluo-N2DH-SIM image with true positives in yellow, false positives in red, and false negatives in green.

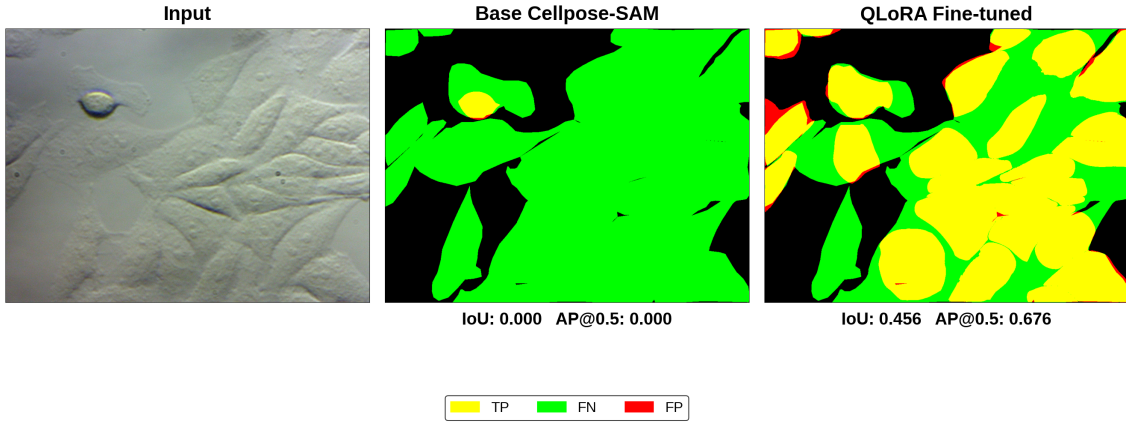


Figure 4.10: Input image, base- and fine-tuned-model segmentation on an image containing uneven lighting from the DIC-HepG2 dataset with true positives in yellow, false positives in red, and false negatives in green. As Cellpose-SAM does not include any contrast enhancement, the base model had a poor performance, but by fine-tuning, the performance, while not perfect, significantly increases.

4.3.4 Out-of-distribution generalisation (CTC: Fluo-C2DL-MSD)

To assess whether fine-tuning had compromised the model’s generalisation ability, the fine-tuned model was evaluated against the baseline on Fluo-C2DL-MSD. As shown in Table 4.4 and Figure 4.11, improvements are observed across all metrics. AP@0.5 increases from 0.406 to 0.483, AP@0.75 from 0.144 to 0.167, and AP@0.9 more than doubles (0.009 to 0.018). The error rate decreases from 1.349 to 1.173, and instance IoU improves modestly from 0.458 to 0.499. These results suggest that fine-tuning generalises beyond the training distribution rather than causing overfitting to target-domain morphologies.

Table 4.4: Segmentation performance of base- and fine-tuned model on unseen out-of-distribution (OOD) data (CTC: Fluo-C2DL-MSD, $n = 51$). Best result per metric in **bold**.

Model	F1 Error	Error Rate	AP@0.5	AP@0.75	AP@0.9	Inst. IoU
Base Cellpose-SAM	0.4596	1.3487	0.4062	0.1438	0.0088	0.4575
Fine-tuned QLoRA	0.3810	1.1734	0.4827	0.1666	0.0178	0.4994

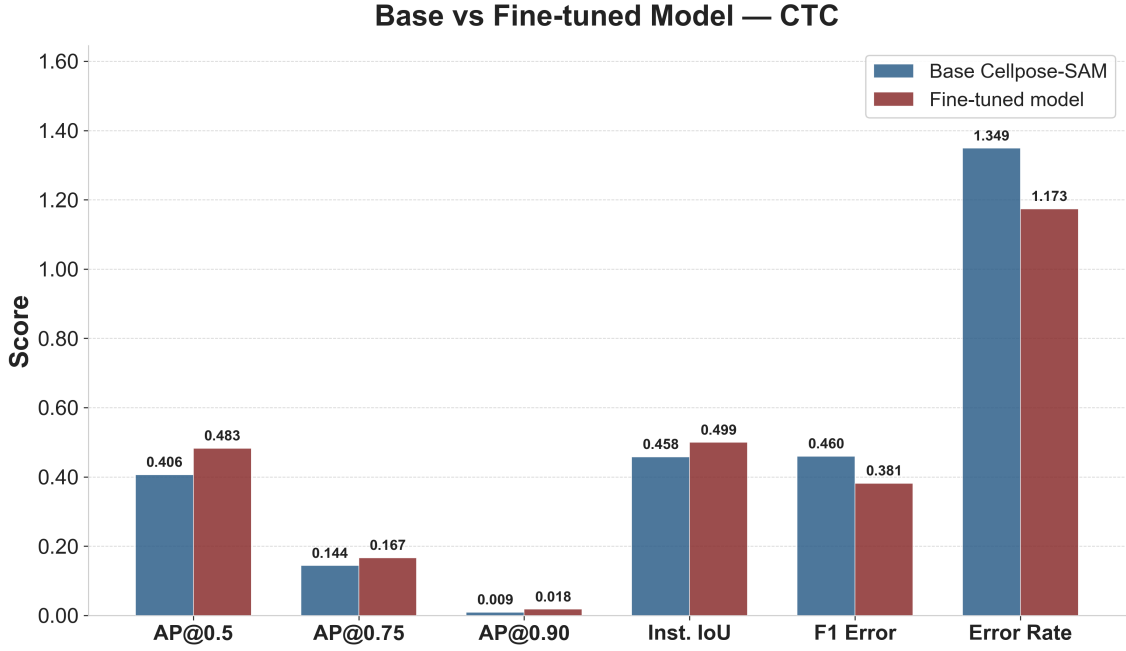


Figure 4.11: Segmentation performance on the unseen out-of-distribution CTC dataset (Fluo-C2DL-MSD, $n = 51$), evaluating generalisation beyond the fine-tuning distribution. Shaded columns indicate lower-is-better metrics.

4.3.5 Domain retention (Cellpose test set)

Finally, to evaluate the boundaries of the fine-tuned model’s domain retention, performance on the original **Cellpose** test set was assessed, which spans a diverse range of cell morphologies. As shown in Table 4.5 and Figure 4.12, the fine-tuned model exhibits a minor regression on F1 error (0.084 to 0.089), error rate (0.160 to 0.168), AP@0.5 (0.855 to 0.847), and instance IoU (0.747 to 0.739), while marginally improving AP@0.75 (0.632 to 0.636) and AP@0.9 (0.167 to 0.175). This marginal decline on highly diverse, out-of-tuning-distribution images highlights the expected trade-off when specialising an FM for specific microscopy morphologies, although the marginal decline across the metrics suggests that QLoRA preserves the majority of the base model’s general-purpose representations.

Table 4.5: Segmentation performance on the **Cellpose** test set. Best result per metric in **bold**.

Model	F1 Error	Error Rate	AP@0.5	AP@0.75	AP@0.9	Inst. IoU
Base Cellpose-SAM	0.0836	0.1602	0.8548	0.6322	0.1669	0.7468
Fine-tuned QLoRA	0.0891	0.1681	0.8465	0.6357	0.1751	0.7387

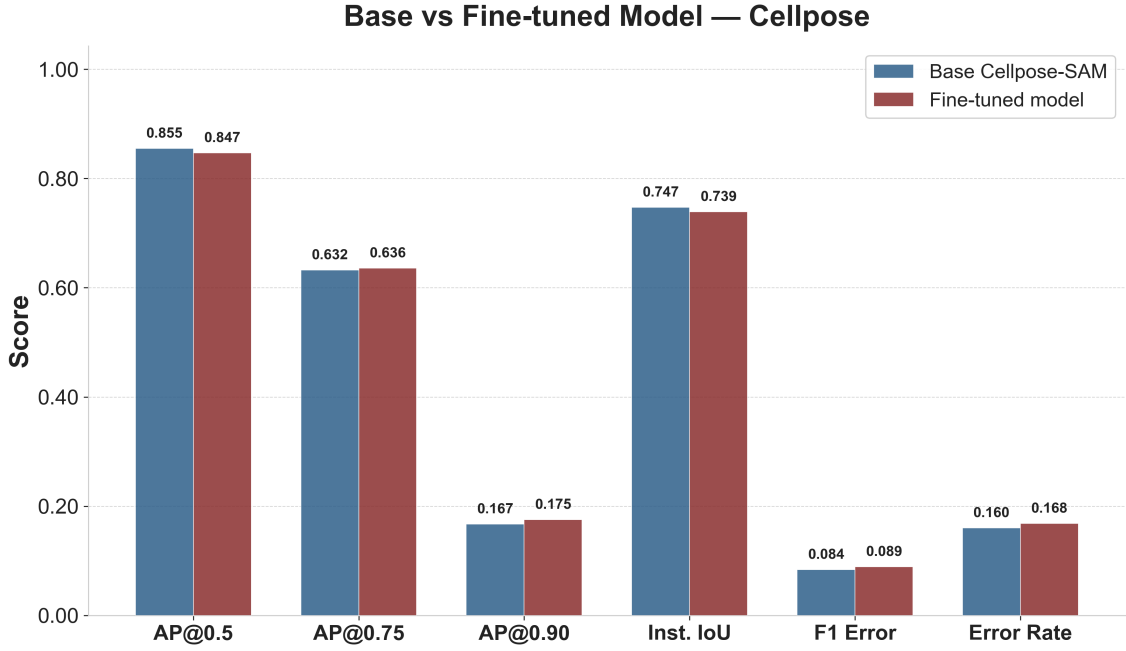


Figure 4.12: Segmentation performance of base- and fine-tuned model on the **Cellpose** test set, used to monitor domain retention. Shaded columns indicate lower-is-better metrics.

4.3.6 Further generalisation evaluation

The evaluation on the **ALFI**, **Phase400**, and **StarDistBF** datasets is shown in Table D.1 - D.3. Two of the three datasets carry annotation limitations that complicate direct metric interpretation. The ground truth masks from the **ALFI** dataset contain cell nuclei only, meaning whole-cell predictions are penalised as false positives across AP thresholds. Instance IoU is less affected, as long as the predicted mask centroid falls within the nucleus ground truth region. This is reflected in the results: despite a modest instance IoU gain, AP metrics decline, and the error rate increases substantially. **StarDistBF** presents a similar issue, where background cells are unannotated, so segmenting them is penalised regardless of quality. The marginal iIoU gain alongside declining AP and a higher error rate is again consistent with the model producing valid but unannotated detections rather than genuine performance degradation.

This interpretation is supported qualitatively by Figures 4.13 and 4.14. On the **ALFI** set, the fine-tuned model produces markedly more pronounced whole-cell activations compared to the nucleus-centered representations of the base model. On some elongated cells, the flow gradients do not fully converge to a clean centre point, which may reflect competing gradients between the nucleus and the whole-cell centroid, compounded by punctate nuclear structure and elongated cell shape. Despite this, the fine-tuned model detects more cells overall, suggesting the representational shift toward whole-cell activation is genuine even if flow convergence is imperfect on atypical morphologies. On **StarDistBF**, the fine-tuned model produces sharper inter-cellular boundaries in the probability map (Figure 4.14), and detects

background cells absent from the ground truth annotations. These valid but unannotated detections register as false positives, consistent with the declining AP and higher error rate (+0.231).

The CellSAM Phase400 dataset uses whole-cell annotations and is therefore the most interpretable of the three sets. AP@0.5 and instance IoU are near-identical between base and fine-tuned model (+0.006 and +0.002, respectively), while AP@0.75 and AP@0.9 show minor declines (-0.036 and -0.016). Overall, differences are small in magnitude and do not indicate systematic degradation on unseen PhC data with clean annotations.

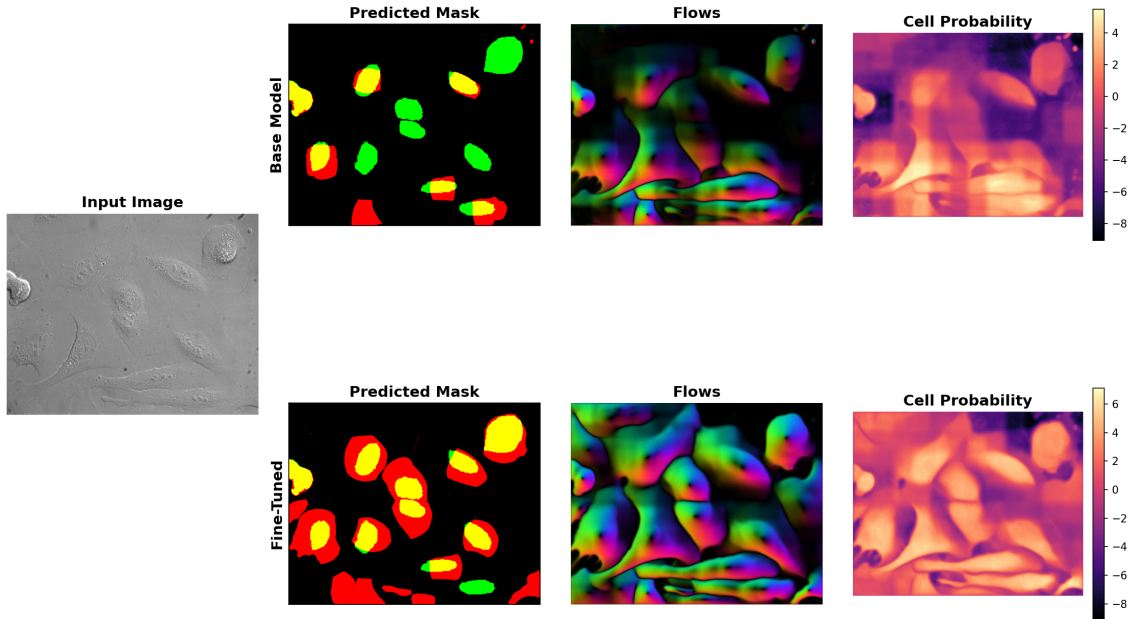


Figure 4.13: Comparison of segmentation mask, gradient flows, and cell probability map for base- (upper) and fine-tuned (lower) model on an image from the ALFI dataset. While the image’s ground truth mask only contains nuclei, making metric comparison infeasible, one can see in the figure that the fine-tuned model is more successful in finding the cells compared to the base model. In the predicted masks, true positives are shown in yellow, false positives in red, and false negatives in green.

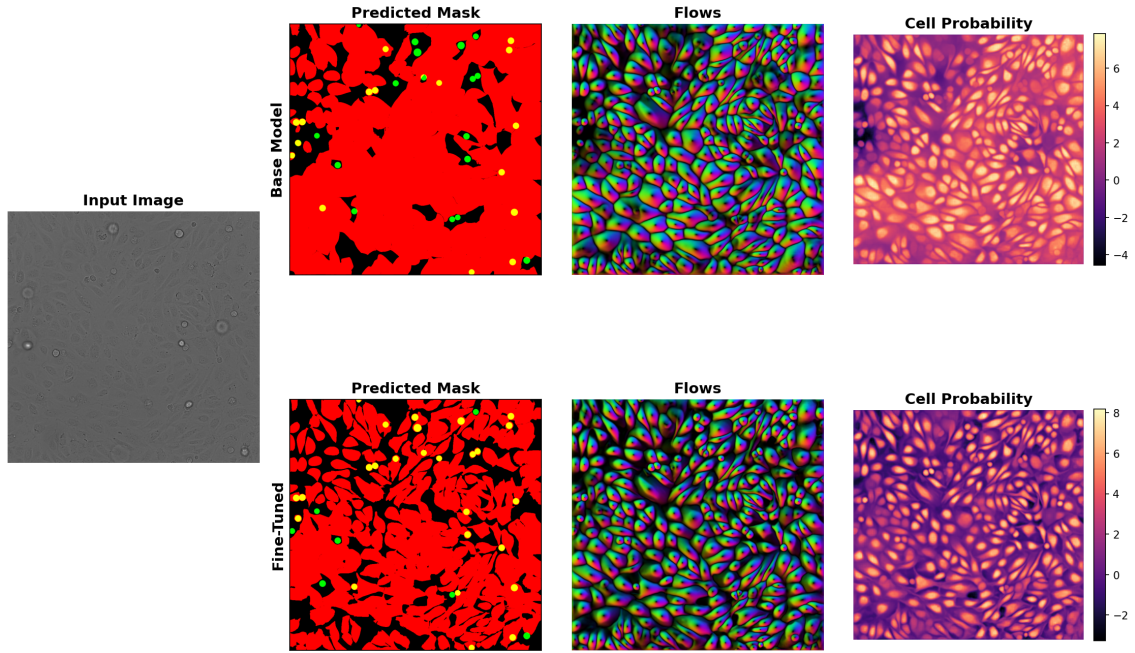


Figure 4.14: Comparison of segmentation mask, flows, and cell probability between base- (upper) and fine-tuned model (lower) on an image from the `StarDist_BF` dataset. As the image contains a background of cells not included in the ground truth mask, direct metric comparison does not give any important insight. However, one can see that the fine-tuned model creates more accurate segmentations of these background cells, where it can, to a higher degree, distinguish actual background from the background cells. In the predicted mask, true positives are shown in yellow, false positives in red, and false negatives in green.

4.4 Discussion

The results demonstrate that QLoRA-based fine-tuning of Cellpose-SAM on a modest curated dataset yields substantial performance gains on the target domain while largely preserving the base model’s general-purpose capabilities. The following subsections discuss the implications of the hyperparameter sweep, the qualitative changes observed in the model’s learned representations, and the degree to which catastrophic forgetting was mitigated.

4.4.1 Fine-tuning strategy and hyperparameter optimisation

The parameter sweep results indicate that the fine-tuning outcome of the training setup is robust across a wide range of configurations, with all five finalists achieving AP@0.5 between 0.747 and 0.778 on the held-out test set. This consistency suggests that the main source of improvement stems from the fine-tuning procedure itself, both the QLoRA encoder adaptation and the unfrozen neck, rather than from precise hyperparameter tuning.

The learning rate sweep revealed a trade-off between in-distribution performance and generalisation. The highest evaluated rate, 5×10^{-4} , achieved the strongest test set performance (AP@0.5 = 0.792, iIoU = 0.730) but at the cost of out-of-distribution generalisation. The **Cellpose** control iIoU dropped from 0.747 to 0.732, and CTC iIoU collapsed to 0.134 against a base model value of 0.458, a sign of catastrophic forgetting caused by a high learning rate and overfitting to the fine-tuning distribution. Conversely, at very low learning rates, weight updates are insufficient for meaningful adaptation, and the fine-tuned model remains within the noise and training-variance of the baseline. For the model setup in the learning rate sweep, the results suggest that a learning rate of 5×10^{-5} is suitable for domain adaptation while retaining out-of-distribution performance.

The block sweep suggests that mid-level representations contribute to domain adaptation, as restricting adapters to the final six blocks (18-23) produced the weakest test-set performance (AP@0.5 = 0.701). Beyond this, the differences between the remaining block configurations were small and likely within training variance, making it difficult to draw strong conclusions about the optimal injection width. Blocks 12-23 were selected for the final configuration as a practical middle-ground, avoiding the added risk of disrupting low-level pre-trained representations associated with full-width injection while covering the mid-to-high level features most relevant to domain shift.

The batch size and rank sweep (Table C.3) showed that reducing the adapter rank from 32 to 16 produced near-identical in-distribution performance (AP@0.5 0.773 vs. 0.771, iIoU 0.714 vs. 0.717) as well as retention (**Cellpose** iIoU: 0.739 vs. 0.734) and an improvement on the CTC set (CTC iIoU: 0.499 vs. 0.483). This suggests that a lower-rank adapter is sufficient for domain adaptation while better preserving the base model’s broader representations. The batch size had limited influence on the detection metrics, though batch size 1 increased test set performance (AP@0.5 = 0.783) at the cost of a notable CTC iIoU drop to 0.371, suggesting training instability

at very small batches.

The combination sweep (Table C.4) explored rank, learning rate, and injection width jointly. The lowest learning rate (1×10^{-5}) produced noticeably weaker test set performance ($\text{AP@0.5} = 0.734$, $\text{iIoU} = 0.657$), consistent with underfitting observed in the LR sweep. Among the remaining configurations, test set differences are small and likely within training variance. As rank, learning rate, and injection width were varied simultaneously, the sweep does not allow clean attribution of differences to individual factors. The configuration with blocks 12-23, $r=16$, $\alpha = 32$, LR 5×10^{-5} was selected based on the strongest CTC generalisation ($\text{iIoU} = 0.515$) and the best **Cellpose** retention ($\text{iIoU} = 0.737$) among the non-underfitting configurations.

Feature distillation produced no consistent improvement over equivalent non-distillation configurations. Across the four matched config pairs, test set AP@0.5 differences ranged from -0.018 to $+0.028$ with no consistent sign, indicating the effect is within training variance rather than a systematic gain. The configuration where distillation appeared strongest (0-23, 16/32, 1×10^{-4} : $\text{AP@0.5} = 0.778$, $\text{iIoU} = 0.716$) also happened to be the best-performing configuration overall, but the same hyperparameters without distillation achieved $\text{AP@0.5} = 0.772$ and $\text{iIoU} = 0.704$. This minor difference suggests that the configuration choice drives performance more than distillation itself. Given that distillation incurs additional overhead (teacher forward pass, approximately 1.5 GB additional GPU memory), the lack of consistent gains does not justify its use under the current setup. As the distillation weight and hook placement were not independently optimised, the approach was likely not fully explored, and dedicated hyperparameter tuning may yield further gains in future work.

Given the marginal performance difference between finalist configurations, the final model selection was guided primarily by parameter efficiency and generalisation stability rather than marginal test-set improvements, favouring the simpler model when performance is comparable. The finalists from the batch/rank sweep and the combination sweep, both using blocks 12-23, $r = 16$, $\alpha = 32$, are the preferred choices, as they adapt only 1.3% of the model’s parameters while achieving strong performance across all three evaluation sets.

4.4.2 Qualitative analysis of fine-tuning result

Examining the flow and cell probability maps reveals how representations evolved during fine-tuning. In Figures 4.15-4.17, the flow and cell probability maps are shown for the base model in the top row and the fine-tuned model in the row below.

In Figure 4.15, the base model clearly struggles to produce meaningful flows or cell probability estimates, yielding diffuse and uninformative representations, which was previously noted in Figures 3.8 and 4.6. In contrast, the fine-tuned model produces clear directional gradients in the flow map alongside a highly detailed and accurate cell probability map. Notably, the fine-tuned model surpasses the base model with the ground truth diameter supplied, improving instance IoU from 0.689 to 0.770 and AP@0.5 from 0.87 to 0.932, demonstrating that fine-tuning has resolved the

scale ambiguity implicitly rather than merely compensating for it. The resulting segmentation masks are shown in Figure 4.6 in Section 4.3.3.

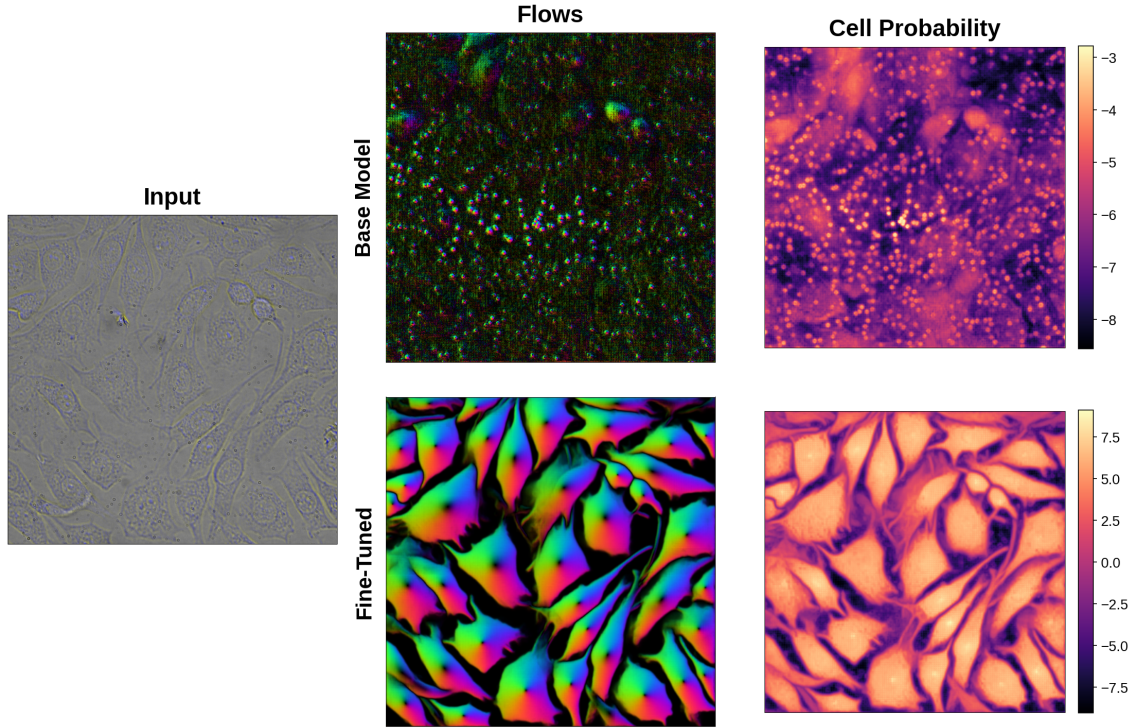


Figure 4.15: Flow and cell probability for base (upper) and fine-tuned (lower) model on an image from **Revvity** containing large out-of-distribution cells. While the base model simply finds background noise, the fine-tuned model can correctly find and segment these large cells.

Figure 4.16 shows a sample previously introduced in Figure 4.7. The cause of the overprediction discussed earlier, which results in a false positive, can be attributed to the higher cell probability and more well-defined flow gradients in the fine-tuned model. It is visible that both models recognise something in the top right corner, but the increased sensitivity of the fine-tuned model causes this to result in a false positive. Notably, this is also what allows the model to identify the cell in the bottom-left of the image. The models have similar intensities in the probability map, yet the fine-tuned model produces considerably more well-defined gradient flows. This leads the fine-tuned model to accurately segment this cell, whereas the base model misses it entirely.

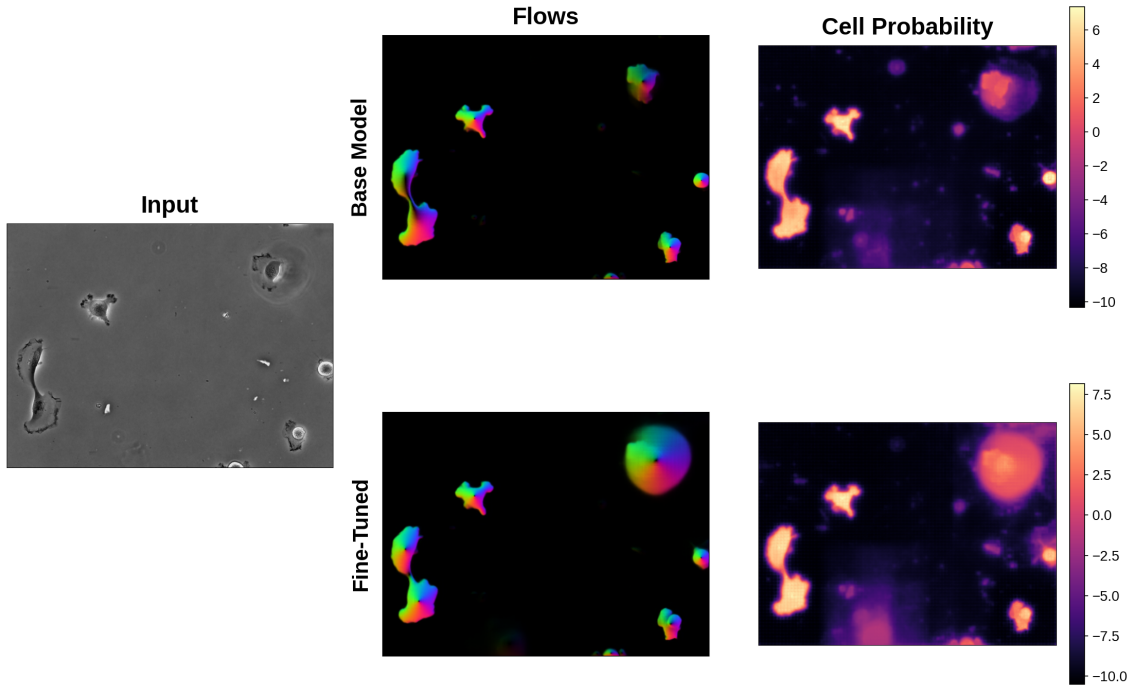


Figure 4.16: Flow and cell probability for base (upper) and fine-tuned (lower) model on an image from the dataset PhC-C2DH. The fine-tuned model produces more accurate flow gradients and detects more details, as seen in the cell in the upper right corner.

Figure 4.17 corresponds to an image on which the base model struggled, as shown in Figure 4.10. The fine-tuned model produces a considerably more detailed cell probability and gradient map, which results in a large improvement in segmentation performance. However, while the cell probability map is more well defined, some gradients on the left side of the image are blurry. This is due to low contrast in this region, which suggests that low-contrast regions continue to pose a challenge.

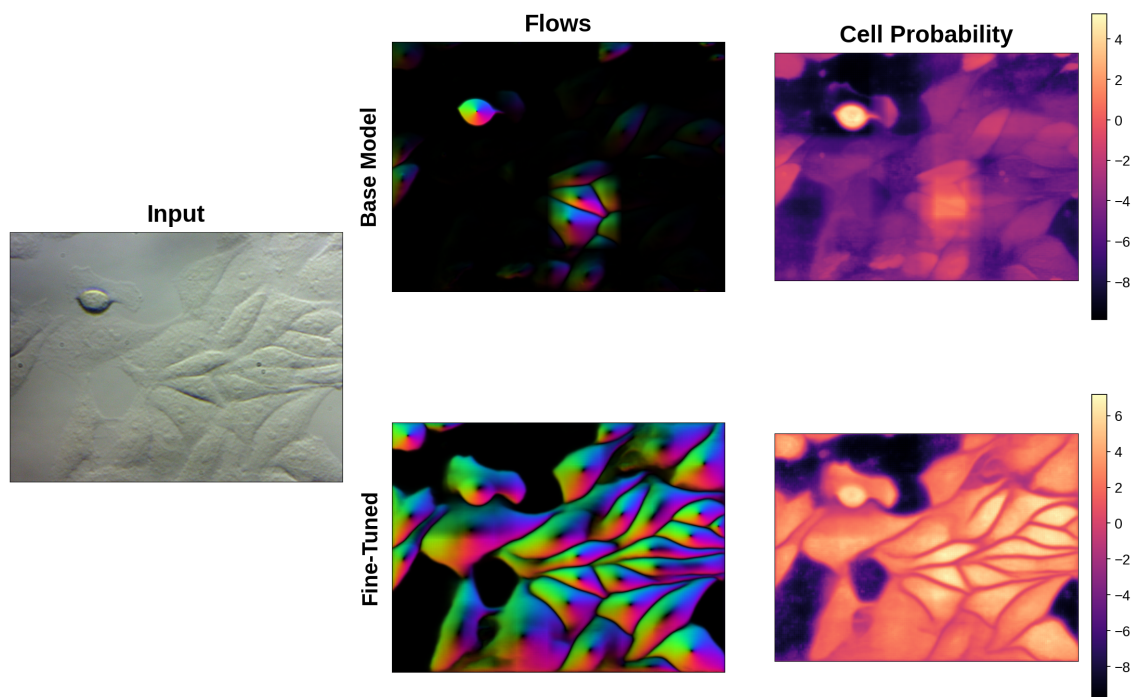


Figure 4.17: Flow and cell probability for base (upper) and fine-tuned (lower) model on an image containing uneven lighting from the DIC-HepG2 dataset. While the base model only finds the solitary cell in the left part of the image, the fine-tuned model finds the majority of cells illustrating the effect of fine-tuning.

4.4.3 Catastrophic forgetting

To ensure that the fine-tuning process did not degrade the model’s baseline capabilities, the fine-tuned model was compared to the base Cellpose-SAM model on two independent datasets: **Fluo-C2DL-MS**C from the CTC and the original **Cellpose** test set. The results indicate that integrating QLoRA with a frozen backbone successfully mitigated catastrophic forgetting when paired with a suitable configuration of learning rate and QLoRA parameters. Feature distillation showed minimal effect on both performance and forgetting prevention. Given that the distillation process incurs additional computational overhead by requiring forward passes through a teacher model, the marginal benefit did not justify the added cost.

Interestingly, on the CTC dataset, the fine-tuned model demonstrated gains over the base model, though the sample size ($n=51$) must be taken into consideration. Conversely, the evaluation on the original **Cellpose** dataset provided insight into the model’s domain boundaries. This dataset contains highly diverse images that differ from the fine-tuning data, including tissues with only green-channel fluorescence and non-microscopy natural objects (e.g., fruits, rocks), intended to test broad generalisability. On these out-of-distribution images, the fine-tuned model performed on par with the base model, with no significant differences observed across metrics.

The three additional generalisation datasets are difficult to interpret as forgetting evidence due to annotation mismatches in two of the three cases. The near-neutral deltas on **Phase400**, which carries clean whole-cell annotations, are the most meaningful signal and confirm that the fine-tuned model retains baseline capability on unseen PhC data. The **ALFI** and **StarDistBF** results are consistent with a representational shift toward whole-cell segmentation rather than forgetting, but this cannot be confirmed without corrected annotations.

5

Conclusion

This thesis evaluated and adapted FMs for automated cell segmentation in microscopy imaging, addressing two connected objectives: benchmarking state-of-the-art models in zero-shot settings and fine-tuning the superior model using parameter-efficient methods.

5.1 Benchmark

The benchmark evaluation across 18 datasets demonstrated that both CellSAM and Cellpose-SAM substantially outperform the traditional StarDist baseline, supporting the broader shift toward FMs for microscopy segmentation. Cellpose-SAM consistently outperformed CellSAM across most datasets and modalities, with the performance gap most pronounced at higher IoU thresholds. This advantage is attributed to Cellpose-SAM’s pixel-level prediction approach, which produces dense flow fields and cell probability maps across the entire image. This results in more precise boundary delineation compared to CellSAM’s prompt-based approach, where segmentation masks are generated from bounding box prompts produced by CellFinder, inherently limiting boundary precision. In dense regions, Cellpose-SAM’s flow-based instance separation further handles touching and overlapping cells more effectively, as each cell’s internal flow pattern converges to a unique center point.

5.2 Fine-tuning

Fine-tuning Cellpose-SAM with QLoRA adapters and an unfrozen neck on a curated dataset of 358 images yielded substantial improvements over the base model on the held-out test set, with AP@0.5 increasing from 0.539 to 0.773 and instance IoU from 0.502 to 0.714. Crucially, these gains did not come at the cost of generalisation: the fine-tuned model improved across all metrics on the unseen CTC out-of-distribution dataset, and exhibited only marginal regression on the Cellpose control set, with AP@0.5 declining by less than 0.01.

The hyperparameter search revealed that adapter injection width is a critical factor, as restricting adapters to the shallowest configuration (blocks 18-23) produced noticeably weaker performance, while deeper injection must be paired with a carefully tuned learning rate to avoid catastrophic forgetting. A lower adapter rank ($r=16$,

$\alpha = 32$) proved sufficient for domain adaptation while better preserving the base model’s broader representations.

Feature distillation did not yield consistent performance gains over equivalent non-distillation configurations. Across the four matched config pairs in Tables C.4 and C.5, test set AP@0.5 differences were small with no consistent sign, suggesting the effect lies within training variance. The configuration that numerically led the distillation sweep (Blocks 0-23, rank = 16, $\alpha = 32$, lr = 1×10^{-4} : AP@0.5 0.778) differed by only 0.006 from its non-distillation counterpart (AP@0.5 0.772), indicating the hyperparameter configuration drives performance more than distillation itself. As λ and hook placement were not independently optimised, the approach was likely not fully explored, and a dedicated sweep remains a natural direction for future work. Given that distillation also introduces non-trivial overhead: a full teacher forward pass per batch, and approximately 1.5 GB of additional GPU memory, the lack of consistent gains does not justify its use under the current setup.

The selected configuration, blocks 12-23, r=16, $\alpha = 32$, lr = 5×10^{-5} , adapts only 1.3% of the total model’s parameters while achieving the strongest generalisation balance across all three evaluation sets, making it the recommended configuration for adapting Cellpose-SAM to new microscopy domains.

Qualitative analysis of the learned representations further supports the effectiveness of fine-tuning. On the Revvity dataset, where the base model produced diffuse and uninformative flow fields, the fine-tuned model learned structured directional gradients and accurate cell probability maps. This improved instance IoU from 0.689 to 0.770 even when the base model was provided with ground truth cell diameter, demonstrating that the performance gain reflects genuine domain adaptation rather than a correction for scale miscalibration alone. However, low contrast regions continue to pose a challenge, with gradient flows remaining blurry where image contrast is insufficient, suggesting that contrast-aware training data or preprocessing remains a direction for future work.

6

Future work

More balanced fine-tuning data: The fine-tuning dataset was dominated by two sources, DynamicNuclearNet and Revvity, which together account for the majority of training images. Despite this imbalance, the largest per-dataset gains were not simply proportional to representation. Revvity showed the greatest improvement, while DynamicNuclearNet sat mid-table, suggesting that data volume alone does not determine where the model improves most, and that the difficulty and visual diversity of a dataset matter more than its share of training examples. A more uniformly balanced dataset, or one constructed by weighting each dataset’s loss contribution inversely to its size, would allow a cleaner assessment of whether gains generalise across imaging conditions and would reduce the risk that smaller, underrepresented datasets are systematically underfitted.

Automated hyperparameter search: The sequential sweep strategy used in this thesis was computationally efficient but did not guarantee an optimal solution. The high dimensionality of the hyperparameter space, covering QLoRA parameters, learning rate, injection width, and distillation parameters, means that global optima outside the explored regions cannot be ruled out. Thus, utilising an automated hyperparameter search across the hyperparameter space, rather than the current sequential narrowing, would likely yield better configurations.

Wider evaluation domain: As the fine-tuned model was evaluated on one dataset per imaging modality, it did not capture the whole domain that Cellpose-SAM was trained on. Fields such as histopathology were omitted as the aim was to segment cells. However, as the models were initially trained on histopathology images, it would be interesting to analyse performance on such images to determine if any catastrophic forgetting occurred.

Bibliography

- [1] E. Moen, D. Bannon, T. Kudo, W. Graf, M. Covert, and D. Van Valen, “Deep learning for cellular image analysis,” *Nature Methods*, vol. 16, no. 12, pp. 1233–1246, 2019. DOI: 10.1038/s41592-019-0403-1.
- [2] M. Maka, V. Ulman, P. Delgado-Rodriguez, et al., “The cell tracking challenge: 10 years of objective benchmarking,” *Nature Methods*, vol. 20, no. 7, pp. 1010–1020, 2023. DOI: 10.1038/s41592-023-01879-y. [Online]. Available: <https://doi.org/10.1038/s41592-023-01879-y>.
- [3] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Springer, 2015, pp. 234–241. [Online]. Available: <https://arxiv.org/abs/1505.04597>.
- [4] 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. DOI: 10.1038/s41592-020-01018-x.
- [5] R. Bommasani et al., *On the opportunities and risks of foundation models*, 2022. arXiv: 2108.07258 [cs.LG]. [Online]. Available: <https://arxiv.org/abs/2108.07258>.
- [6] M. Marks et al., “Cellsam: A foundation model for cell segmentation,” *Nature Methods*, vol. 22, no. 12, pp. 2585–2593, 2025. DOI: 10.1038/s41592-025-02879-w. [Online]. Available: <https://doi.org/10.1038/s41592-025-02879-w>.
- [7] M. Pachitariu, M. Rariden, and C. Stringer, “Cellpose-sam: Superhuman generalization for cellular segmentation,” *bioRxiv*, 2025. DOI: 10.1101/2025.04.28.651001. [Online]. Available: <https://www.biorxiv.org/content/early/2025/05/01/2025.04.28.651001>.
- [8] E. J. Hu et al. “LoRA: Low-Rank Adaptation of Large Language Models.” arXiv: 2106.09685 [cs.CL]. [Online]. Available: <https://arxiv.org/abs/2106.09685>.
- [9] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer. “QLoRA: Efficient Finetuning of Quantized LLMs.” arXiv: 2305.14314 [cs.LG]. [Online]. Available: <https://arxiv.org/abs/2305.14314>.
- [10] M. Bickle, “The beautiful cell: High-content screening in drug discovery,” *Analytical and Bioanalytical Chemistry*, vol. 398, no. 1, pp. 219–226, 2010. DOI: 10.1007/s00216-010-3788-3.
- [11] A. Archit et al., “Segment anything for microscopy,” *Nature Methods*, vol. 22, no. 3, pp. 579–591, 2025. DOI: 10.1038/s41592-024-02580-4.

- [12] J. Ma et al., “The multimodality cell segmentation challenge: Toward universal solutions,” *Nature Methods*, vol. 21, no. 6, pp. 1103–1113, 2024. DOI: 10.1038/s41592-024-02233-6.
- [13] M. Schwendy, R. E. Unger, and S. H. Parekh, “EVICAN—a balanced dataset for algorithm development in cell and nucleus segmentation,” *Bioinformatics*, vol. 36, no. 12, pp. 3863–3870, 2020. DOI: 10.1093/bioinformatics/btaa225.
- [14] J. W. Lichtman and J.-A. Conchello, “Fluorescence microscopy,” *Nature Methods*, vol. 2, no. 12, pp. 910–919, 2005. DOI: 10.1038/nmeth817.
- [15] G. Wang and N. Fang, “Detecting and tracking nonfluorescent nanoparticle probes in live cells,” in *Methods in Enzymology*, P. M. Conn, Ed., vol. 504, Academic Press, 2012, pp. 83–108, ISBN: 978-0-12-391857-4. DOI: 10.1016/B978-0-12-391857-4.00004-5.
- [16] R. W. Lovitt and C. J. Wright, “Microscopy | light microscopy,” in *Encyclopedia of Food Microbiology*, R. K. Robinson, Ed., Elsevier, 1999, pp. 1379–1388, ISBN: 978-0-12-227070-3. DOI: 10.1006/rwfm.1999.1060.
- [17] H. Heydarian, P. Yazdanfar, A. Zarif, and B. Rashidian, “Near field differential interference contrast microscopy,” *Scientific Reports*, vol. 10, no. 1, p. 9644, 2020. DOI: 10.1038/s41598-020-66482-z.
- [18] A. Dosovitskiy et al., *An image is worth 16x16 words: Transformers for image recognition at scale*, 2021. arXiv: 2010.11929 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/2010.11929>.
- [19] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked autoencoders are scalable vision learners,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2022, pp. 16 000–16 009. DOI: 10.1109/cvpr52688.2022.01553.
- [20] A. Vaswani et al., *Attention is all you need*, 2017. arXiv: 1706.03762 [cs.CL]. [Online]. Available: <https://arxiv.org/abs/1706.03762>.
- [21] S. Zheng et al., *Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers*, 2021. arXiv: 2012.15840 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/2012.15840>.
- [22] A. Kirillov et al., *Segment anything*, 2023. arXiv: 2304.02643 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/2304.02643>.
- [23] 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, 2022, pp. 2567–2575. DOI: 10.1609/aaai.v36i3.20158.
- [24] 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*, Springer International Publishing, 2018, pp. 265–273. DOI: 10.1007/978-3-030-00934-2_30. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-00934-2_30.
- [25] M. S. Schwartz et al., “Caliban: Accurate cell tracking and lineage construction in live-cell imaging experiments with deep learning,” *bioRxiv*, 2023. DOI: 10.1101/803205. [Online]. Available: <https://www.biorxiv.org/content/10.1101/803205v4>.

- [26] C. Shi et al., “Cellbindb: A large-scale multimodal annotated dataset for cell segmentation with benchmarking of universal models,” *GigaScience*, vol. 14, giaf069, Jan. 2025. DOI: 10.1093/gigascience/giaf069. [Online]. Available: <https://doi.org/10.1093/gigascience/giaf069>.
- [27] S. K. Rashed, M. Arvidsson, R. Ahmed, and S. Aits, “An annotated high-content fluorescence microscopy dataset with EGFP-Galectin-3-stained cells and manually labelled outlines,” *Data in Brief*, vol. 58, p. 111148, 2025. DOI: 10.1016/j.dib.2024.111148.
- [28] C. Reich, T. Prangemeier, and H. Koeppl, “The tyc dataset for understanding instance-level semantics and motions of cells in microstructures,” in *2023 IEEE/CVF International Conference on Computer Vision Workshops (IC-CVW)*, 2023, pp. 3942–3953. DOI: 10.1109/ICCVW60793.2023.00426.
- [29] Y. Prytula, I. Tsiporenko, A. Zeynalli, and D. Fishman, *Iaunet: Instance-aware u-net*, 2025. arXiv: 2508.01928 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/2508.01928>.
- [30] F. Pan, *Differential interference contrast (dic) image of unstained living hepg2 human liver cancer cells*, 2024. DOI: 10.5281/zenodo.13120679. [Online]. Available: <https://doi.org/10.5281/zenodo.13120679>.
- [31] C. Zhang, F. Huber, M. Knop, and F. A. Hamprecht, “Yeast cell detection and segmentation in bright field microscopy,” in *2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI)*, 2014, pp. 1267–1270. DOI: 10.1109/ISBI.2014.6868107.
- [32] scikit-image developers, *Watershed segmentation - scikit-image 0.26.0 documentation*, Accessed: 2026-02-15, 2024. [Online]. Available: https://scikit-image.org/docs/stable/auto_examples/segmentation/plot_watershed.html.
- [33] D. Liu, M. Kan, S. Shan, and X. Chen, *Function-consistent feature distillation*, 2023. arXiv: 2304.11832 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/2304.11832>.
- [34] L. Antonelli et al., “ALFI: Cell cycle phenotype annotations of label-free time-lapse imaging data from cultured human cells,” *Sci. Data*, vol. 10, no. 1, p. 677, 2023. DOI: 10.1038/s41597-023-02540-1.
- [35] G. Follain, S. Ghimire, J. Pylvänäinen, J. Ivaska, and G. Jacquemet, *Stardist_bf_monocytes_data*, Zenodo, 2024. DOI: 10.5281/zenodo.10572200. [Online]. Available: <https://doi.org/10.5281/zenodo.10572200>.

A

LoRA equations

For a network with i sequential linear layers augmented with LoRA, define the effective weight matrix of layer k as

$$\tilde{W}^{(k)} = W_0^{(k)} + \frac{\alpha_k}{r_k} B^{(k)} A^{(k)}. \quad (\text{A.1})$$

Starting from the input $h^{(0)} = x$, each layer applies its effective weight matrix to the previous activation:

$$h^{(k)} = \tilde{W}^{(k)} h^{(k-1)}, \quad k = 1, \dots, i. \quad (\text{A.2})$$

Thus, the final output after i layers is

$$h^{(i)} = \tilde{W}^{(i)} \tilde{W}^{(i-1)} \dots \tilde{W}^{(1)} x. \quad (\text{A.3})$$

B

Benchmark Results

Table B.1: Mean segmentation metrics by model across all datasets, best result per metric in **bold**

Model	Inst. IoU	AP@0.5	AP@0.75	AP@0.9	F1 Error	Error Rate
Cellpose-SAM	0.594	0.619	0.400	0.146	0.301	0.474
CellSAM	0.537	0.546	0.297	0.052	0.364	0.925
StarDist	0.405	0.316	0.177	0.073	0.607	2.478

Table B.2: Benchmark segmentation results across datasets and modalities. Best result per dataset per metric shown in **bold**, with N indicating the number of images used from the dataset.

Dataset	Model	Inst.IoU	AP@0.5	AP@0.75	AP@0.9	F1Error	ErrorRate	N
FLUORESCENT								
FLUO-C2DL-Huh7	CellSAM	0.754	0.728	0.531	0.101	0.159	0.345	13
	Cellpose-SAM	0.398	0.197	0.038	0.015	0.676	1.505	
	StarDist	0.159	0.004	0.000	0.000	0.993	2.718	
FLUO-N2DH-GOWT1	Cellpose-SAM	0.927	0.904	0.881	0.798	0.052	0.111	184
	CellSAM	0.841	0.828	0.775	0.203	0.097	0.219	
	StarDist	0.903	0.819	0.773	0.545	0.102	0.231	
Fluo-N2DH-SIM	Cellpose-SAM	0.867	0.940	0.853	0.511	0.032	0.063	65
	CellSAM	0.855	0.942	0.836	0.252	0.031	0.062	
	StarDist	0.692	0.550	0.391	0.148	0.306	0.763	
Fluo-N2DL-HeLa-02	Cellpose-SAM	0.880	0.836	0.781	0.573	0.089	0.188	92
	CellSAM	0.856	0.833	0.816	0.214	0.091	0.194	
	StarDist	0.874	0.844	0.770	0.617	0.085	0.177	
DynamicNuclearNet	CellSAM	0.735	0.910	0.447	0.026	0.050	0.103	717
	Cellpose-SAM	0.655	0.856	0.297	0.004	0.082	0.157	
	StarDist	0.619	0.799	0.294	0.001	0.117	0.226	
Aitslab_bioimaging2	Cellpose-SAM	0.679	0.732	0.409	0.044	0.159	0.315	12
	CellSAM	0.650	0.720	0.389	0.038	0.167	0.338	
	StarDist	0.373	0.219	0.042	0.000	0.659	1.505	
CellbinDB_DAPI	Cellpose-SAM	0.606	0.721	0.310	0.041	0.167	0.312	203
	CellSAM	0.584	0.618	0.159	0.018	0.250	0.523	
	StarDist	0.502	0.508	0.164	0.013	0.341	0.732	
CellbinDB_10xgen_DAPI	CellSAM	0.602	0.630	0.237	0.028	0.232	0.468	100
	Cellpose-SAM	0.579	0.675	0.230	0.013	0.205	0.384	
	StarDist	0.571	0.694	0.306	0.021	0.186	0.339	

(continued on next page)

CellbinDB_ssDNA	StarDist	0.514	0.543	0.226	0.031	0.303	0.471	276
	Cellpose-SAM	0.513	0.593	0.276	0.037	0.267	0.471	
	CellSAM	0.471	0.467	0.165	0.020	0.386	0.737	
BRIGHTFIELD								
EVICAN_easy	Cellpose-SAM	0.599	0.685	0.546	0.171	0.255	0.382	33
	CellSAM	0.508	0.466	0.208	0.047	0.462	1.401	
	StarDist	0.142	0.022	0.001	0.000	0.959	4.763	
EVICAN_medium	Cellpose-SAM	0.260	0.319	0.100	0.000	0.615	0.774	33
	CellSAM	0.319	0.213	0.048	0.010	0.705	2.030	
	StarDist	0.084	0.032	0.002	0.000	0.950	5.280	
EVICAN_difficult	Cellpose-SAM	0.183	0.202	0.093	0.004	0.728	0.867	32
	CellSAM	0.234	0.188	0.031	0.000	0.748	1.894	
	StarDist	0.071	0.022	0.007	0.000	0.961	4.950	
TYC_test	Cellpose-SAM	0.741	0.874	0.511	0.035	0.069	0.137	8
	CellSAM	0.696	0.822	0.432	0.015	0.101	0.196	
	StarDist	0.539	0.274	0.169	0.017	0.612	2.932	
TYC_val	Cellpose-SAM	0.719	0.838	0.430	0.040	0.090	0.178	8
	CellSAM	0.666	0.803	0.366	0.020	0.110	0.219	
	StarDist	0.527	0.234	0.137	0.014	0.651	3.401	
Yeast BF	StarDist	0.754	0.408	0.344	0.129	0.421	1.275	15
	Cellpose-SAM	0.673	0.622	0.308	0.002	0.234	0.536	
	CellSAM	0.643	0.610	0.187	0.001	0.243	0.551	
BF-C2DL-HSC	Cellpose-SAM	0.760	0.612	0.500	0.062	0.268	0.699	49
	StarDist	0.290	0.003	0.000	0.000	0.994	10.827	
	CellSAM	0.107	0.128	0.022	0.000	0.832	1.620	
Revvity-25	CellSAM	0.025	0.025	0.005	0.000	0.995	1.000	110
	Cellpose-SAM	0.003	0.003	0.002	0.000	0.995	1.000	
	StarDist	0.002	0.002	0.000	0.000	0.995	2.180	

(continued on next page)

DIC								
HepG2-Cells	CellSAM	0.202	0.239	0.085	0.001	0.652	0.886	104
	StarDist	0.014	0.000	0.000	0.000	1.000	2.617	
	Cellpose-SAM	0.013	0.016	0.006	0.000	0.971	1.015	
DIC-C2DH-HeLa	Cellpose-SAM	0.748	0.779	0.472	0.039	0.132	0.310	17
	CellSAM	0.384	0.253	0.042	0.011	0.662	1.200	
	StarDist	0.088	0.060	0.025	0.002	0.891	1.791	
PHASE CONTRAST								
PhC-C2DH-U373	Cellpose-SAM	0.877	0.816	0.758	0.621	0.116	0.282	34
	CellSAM	0.599	0.452	0.301	0.084	0.408	0.990	
	StarDist	0.253	0.048	0.002	0.000	0.913	4.107	
PhC-C2DL-PSC	Cellpose-SAM	0.788	0.787	0.596	0.050	0.120	0.264	600
	CellSAM	0.547	0.598	0.157	0.003	0.261	0.511	
	StarDist	0.531	0.542	0.057	0.001	0.299	0.640	

C

Parameter Sweep Results

Table C.1: Learning rate sweep results (blocks 12–23, rank 32, $\alpha = 64$, batch size 8). Best result per metric in **bold**.

LR	AP@0.5	AP@0.75	AP@0.9	Inst. IoU	F1 Err	Err Rate
<i>Test set</i>						
Base model	0.539	0.324	0.109	0.502	0.396	0.856
1×10^{-6}	0.550	0.337	0.108	0.508	0.382	0.841
5×10^{-6}	0.692	0.407	0.109	0.611	0.227	0.690
1×10^{-5}	0.719	0.443	0.119	0.634	0.202	0.694
5×10^{-5}	0.760	0.560	0.222	0.710	0.161	0.673
5×10^{-4}	0.792	0.613	0.294	0.730	0.140	0.268
<i>Cellpose control set</i>						
Base model	0.855	0.632	0.167	0.747	0.084	0.160
1×10^{-6}	0.855	0.633	0.164	0.746	0.083	0.159
5×10^{-6}	0.854	0.636	0.167	0.743	0.085	0.160
1×10^{-5}	0.850	0.634	0.157	0.736	0.089	0.164
5×10^{-5}	0.853	0.637	0.166	0.742	0.085	0.162
5×10^{-4}	0.835	0.607	0.163	0.732	0.096	0.183
<i>CTC out-of-distribution</i>						
Base model	0.406	0.144	0.009	0.458	0.460	1.349
1×10^{-6}	0.400	0.135	0.009	0.463	0.462	1.326
5×10^{-6}	0.409	0.156	0.019	0.496	0.451	1.384
1×10^{-5}	0.413	0.158	0.023	0.502	0.445	1.385
5×10^{-5}	0.439	0.154	0.021	0.507	0.421	1.325
5×10^{-4}	0.150	0.060	0.007	0.134	0.779	1.136

Table C.2: Block injection sweep results (rank 32, $\alpha = 64$, LR 4×10^{-5} , batch size 8). Best result per metric in **bold**.

Blocks	AP@0.5	AP@0.75	AP@0.9	Inst. IoU	F1 Err	Err Rate
<i>Test set</i>						
Base model	0.539	0.324	0.109	0.502	0.396	0.856
0–23 (all)	0.770	0.583	0.249	0.704	0.155	0.596
0–23 (r64)	0.773	0.584	0.241	0.697	0.152	0.445
0–11	0.745	0.539	0.222	0.682	0.172	0.655
12–23	0.749	0.550	0.203	0.703	0.167	0.685
18–23	0.701	0.481	0.134	0.654	0.216	0.738
<i>Cellpose control set</i>						
Base model	0.855	0.632	0.167	0.747	0.084	0.160
0–23 (all)	0.841	0.623	0.164	0.728	0.094	0.173
0–23 (r64)	0.839	0.618	0.160	0.726	0.095	0.174
0–11	0.839	0.613	0.150	0.725	0.094	0.174
12–23	0.845	0.633	0.160	0.735	0.092	0.169
18–23	0.831	0.617	0.139	0.718	0.102	0.183
<i>CTC out-of-distribution</i>						
Base model	0.406	0.144	0.009	0.458	0.460	1.349
0–23 (all)	0.510	0.185	0.015	0.459	0.381	1.084
0–23 (r64)	0.448	0.128	0.006	0.406	0.431	1.110
0–11	0.488	0.184	0.013	0.431	0.390	1.107
12–23	0.437	0.158	0.026	0.505	0.425	1.313
18–23	0.405	0.149	0.023	0.493	0.452	1.427

Table C.3: Batch size and rank sweep results (blocks 12–23, LR 1×10^{-4}). Best result per metric in **bold**.

Batch	Rank / α	AP@0.5	AP@0.75	AP@0.9	Inst. IoU	F1 Err	Err Rate
<i>Test set</i>							
Base model	—	0.539	0.324	0.109	0.502	0.396	0.856
8	32 / 64	0.771	0.588	0.239	0.717	0.155	0.643
1	32 / 64	0.783	0.613	0.257	0.720	0.149	0.278
4	32 / 64	0.765	0.577	0.241	0.715	0.159	0.634
8	16 / 32	0.773	0.586	0.228	0.714	0.154	0.657
8	32 / 128	0.770	0.591	0.243	0.715	0.156	0.513
<i>Cellpose control set</i>							
Base model	—	0.855	0.632	0.167	0.747	0.084	0.160
8	32 / 64	0.842	0.623	0.168	0.734	0.092	0.173
1	32 / 64	0.839	0.608	0.160	0.732	0.094	0.177
4	32 / 64	0.839	0.617	0.166	0.729	0.094	0.176
8	16 / 32	0.847	0.636	0.175	0.739	0.089	0.168
8	32 / 128	0.841	0.625	0.172	0.731	0.093	0.173
<i>CTC out-of-distribution</i>							
Base model	—	0.406	0.144	0.009	0.458	0.460	1.349
8	32 / 64	0.441	0.159	0.015	0.483	0.422	1.240
1	32 / 64	0.391	0.115	0.012	0.371	0.489	1.119
4	32 / 64	0.457	0.148	0.015	0.477	0.409	1.152
8	16 / 32	0.483	0.167	0.018	0.499	0.381	1.173
8	32 / 128	0.459	0.155	0.013	0.479	0.408	1.137

Table C.4: Final combination sweep results (batch size 8). Best result per metric in **bold**.

Blocks	Rank / α	LR	AP@0.5	AP@0.75	AP@0.9	Inst. IoU	F1 Err	Err Rate
<i>Test set</i>								
Base model	—	—	0.539	0.324	0.109	0.502	0.396	0.856
0-23	16 / 32	1×10^{-4}	0.772	0.584	0.252	0.704	0.154	0.543
0-23	32 / 64	5×10^{-5}	0.777	0.595	0.252	0.711	0.150	0.573
0-23	32 / 64	1×10^{-5}	0.734	0.516	0.138	0.657	0.186	0.690
12-23	16 / 32	5×10^{-5}	0.747	0.548	0.194	0.694	0.171	0.693
<i>Cellpose control set</i>								
Base model	—	—	0.855	0.632	0.167	0.747	0.084	0.160
0-23	16 / 32	1×10^{-4}	0.835	0.613	0.161	0.728	0.097	0.179
0-23	32 / 64	5×10^{-5}	0.845	0.628	0.161	0.734	0.091	0.170
0-23	32 / 64	1×10^{-5}	0.847	0.626	0.154	0.733	0.090	0.167
12-23	16 / 32	5×10^{-5}	0.852	0.632	0.152	0.737	0.086	0.162
<i>CTC out-of-distribution</i>								
Base model	—	—	0.406	0.144	0.009	0.458	0.460	1.349
0-23	16 / 32	1×10^{-4}	0.473	0.161	0.011	0.456	0.402	1.116
0-23	32 / 64	5×10^{-5}	0.491	0.159	0.012	0.466	0.388	1.107
0-23	32 / 64	1×10^{-5}	0.482	0.188	0.016	0.495	0.392	1.261
12-23	16 / 32	5×10^{-5}	0.438	0.160	0.020	0.515	0.424	1.338

Table C.5: Distillation sweep results (batch size 8). Configurations mirror those in the final combination sweep. Best result per metric in **bold**.

Blocks	Rank / α	LR	AP@0.5	AP@0.75	AP@0.9	Inst. IoU	F1 Err	Err Rate
<i>Test set</i>								
Base model	—	—	0.539	0.324	0.109	0.502	0.396	0.856
0-23	16 / 32	1×10^{-4}	0.778	0.580	0.236	0.716	0.149	0.621
0-23	32 / 64	5×10^{-5}	0.759	0.562	0.224	0.706	0.164	0.685
0-23	32 / 64	1×10^{-5}	0.762	0.571	0.225	0.708	0.161	0.650
12-23	16 / 32	5×10^{-5}	0.753	0.542	0.191	0.687	0.167	0.663
<i>Cellpose control set</i>								
Base model	—	—	0.855	0.632	0.167	0.747	0.084	0.160
0-23	16 / 32	1×10^{-4}	0.849	0.639	0.171	0.738	0.087	0.163
0-23	32 / 64	5×10^{-5}	0.852	0.637	0.168	0.740	0.086	0.162
0-23	32 / 64	1×10^{-5}	0.851	0.640	0.171	0.740	0.087	0.162
12-23	16 / 32	5×10^{-5}	0.850	0.629	0.154	0.737	0.088	0.165
<i>CTC out-of-distribution</i>								
Base model	—	—	0.406	0.144	0.009	0.458	0.460	1.349
0-23	16 / 32	1×10^{-4}	0.468	0.165	0.014	0.494	0.411	1.318
0-23	32 / 64	5×10^{-5}	0.413	0.151	0.023	0.502	0.445	1.423
0-23	32 / 64	1×10^{-5}	0.429	0.155	0.020	0.492	0.434	1.445
12-23	16 / 32	5×10^{-5}	0.422	0.157	0.023	0.510	0.438	1.389

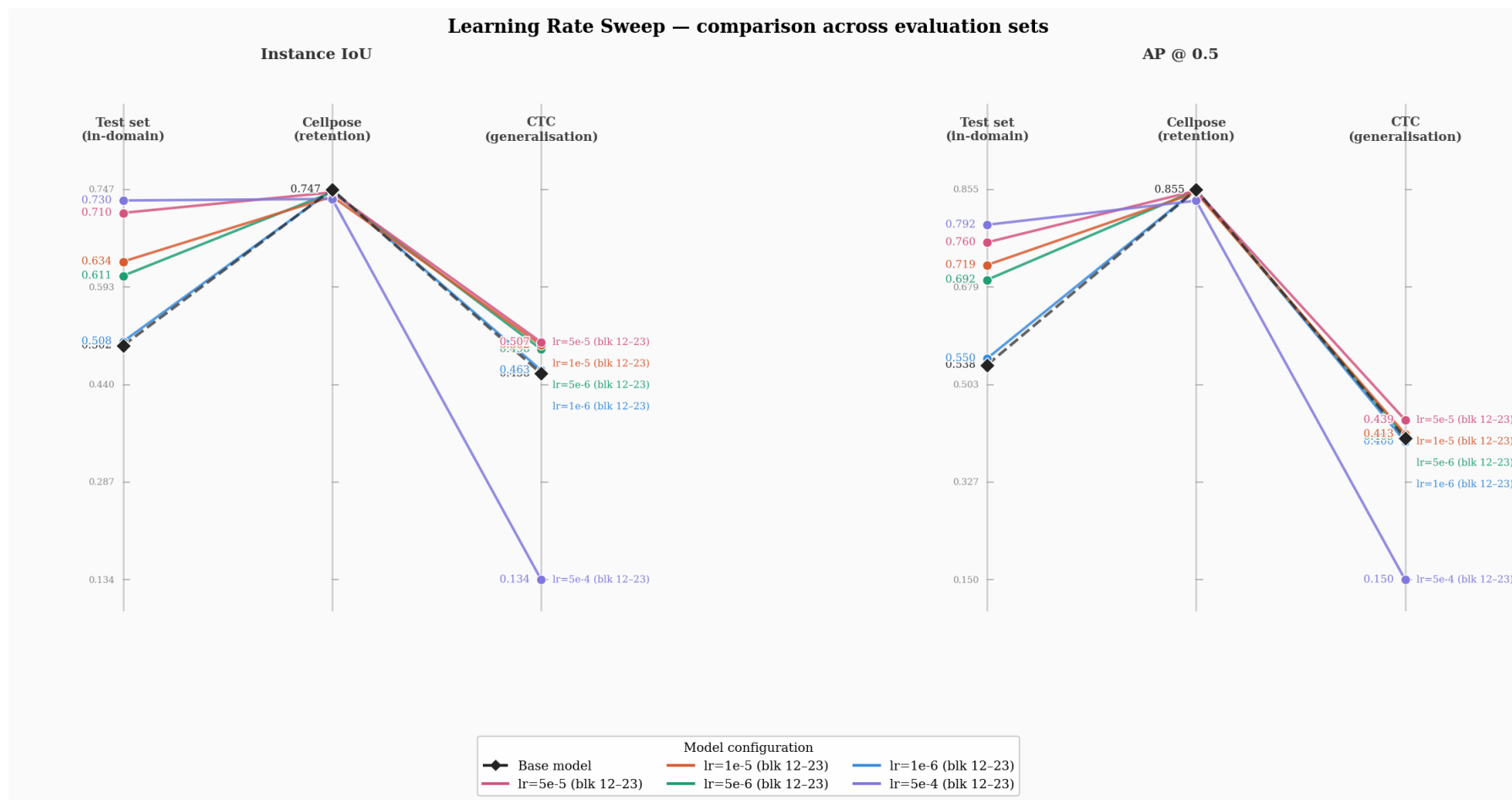


Figure C.1: Comparison of performance across different Learning rates varying from $5e-4$ to $1e-6$. Fine-tuned configurations are shown as colored lines, and the original base model is shown in dashed black.

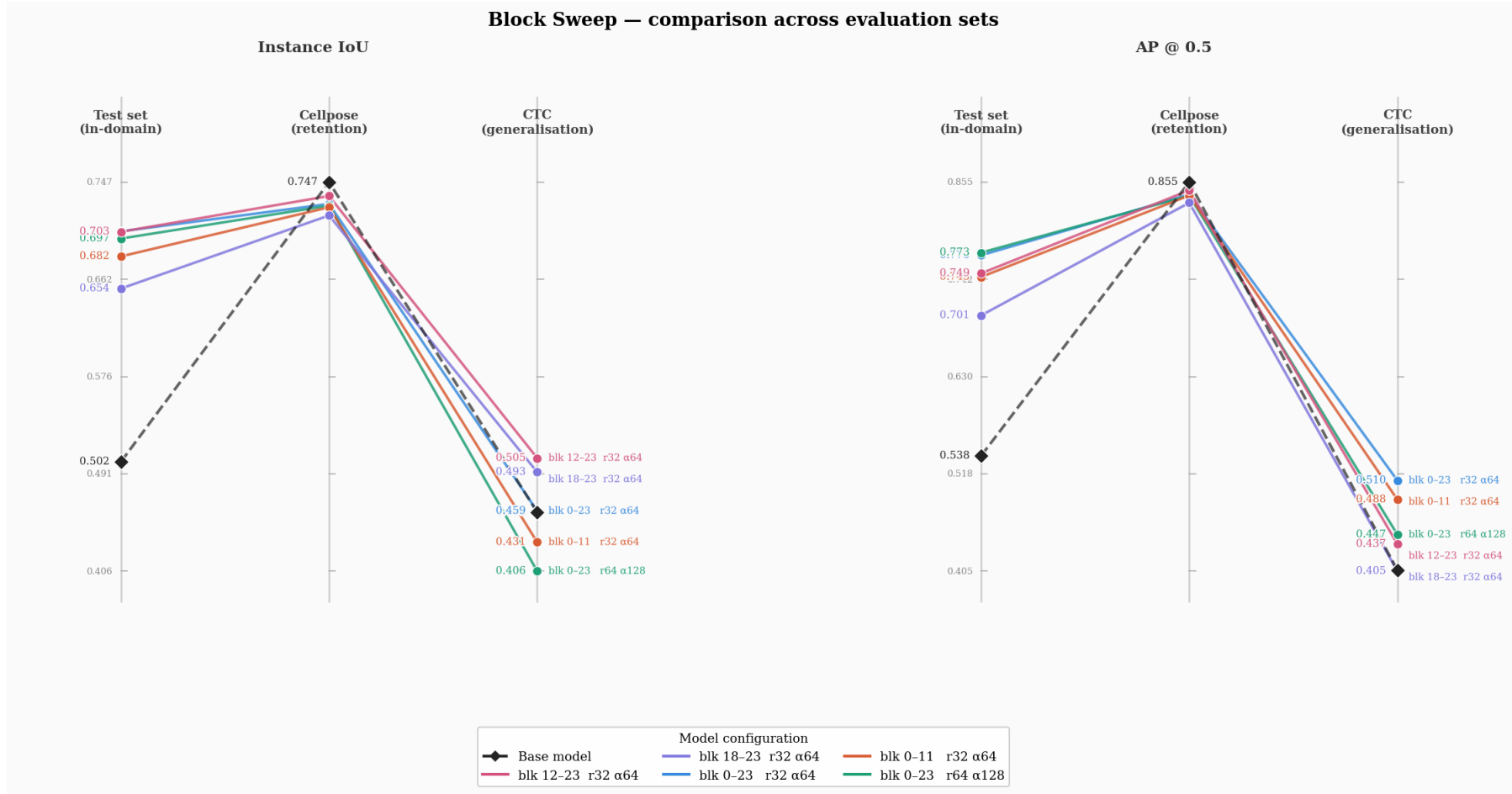


Figure C.2: Comparison of performance when injecting QLoRA into the first half of blocks (blk 0-11), last half (blk 12-23), final blocks (18-23), and all blocks (0-23). A configuration of (rank 32, $\alpha = 64$, LR $5e-5$, batch size 8) was used with the exception of the green configuration, which used a rank of 64, and an $\alpha = 128$

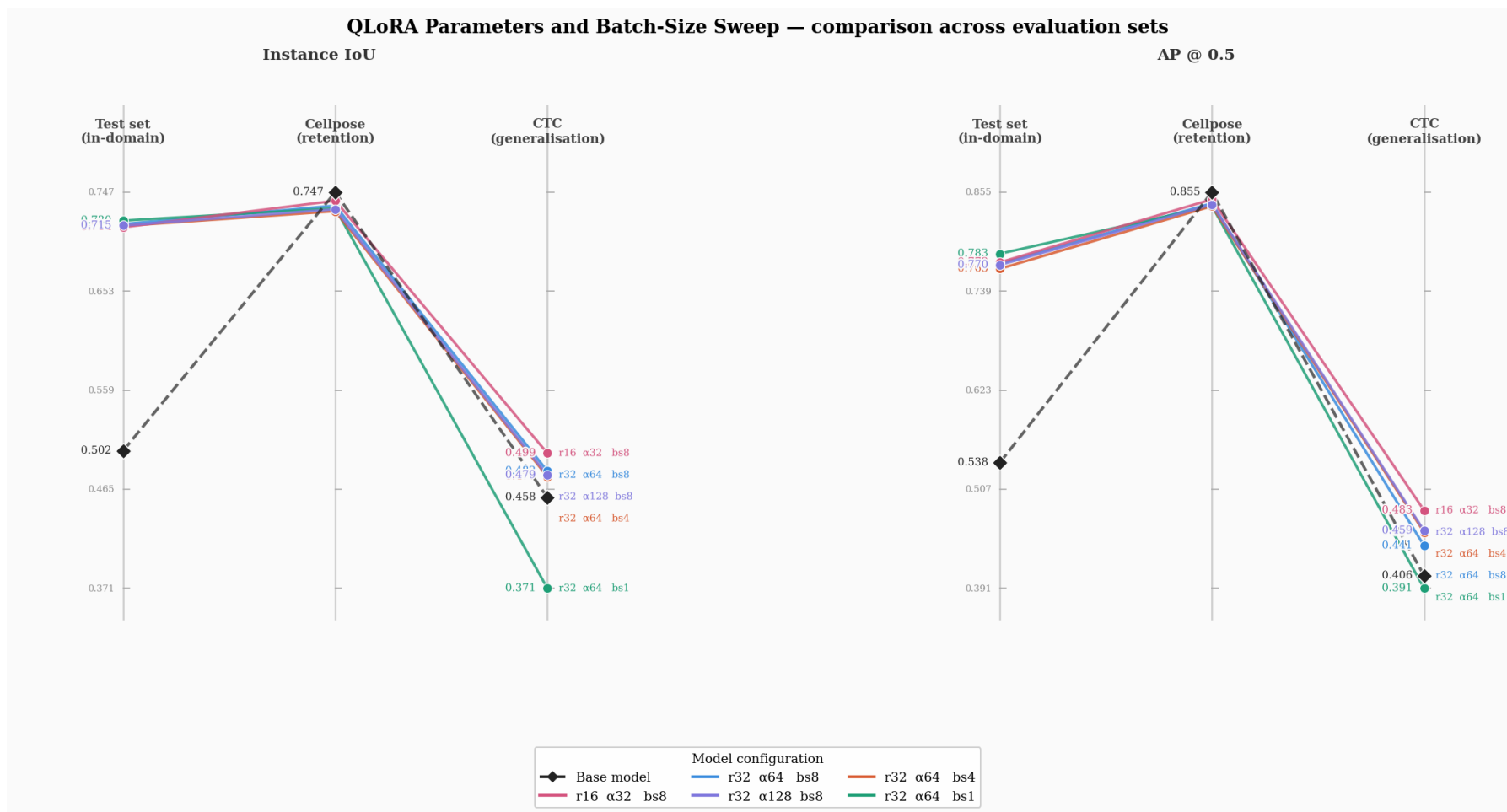


Figure C.3: Comparison of performance with varying QLoRA parameters (rank (16-32), α (32-128), and batch-size (1-8). blocks 12–23, LR 5e-5

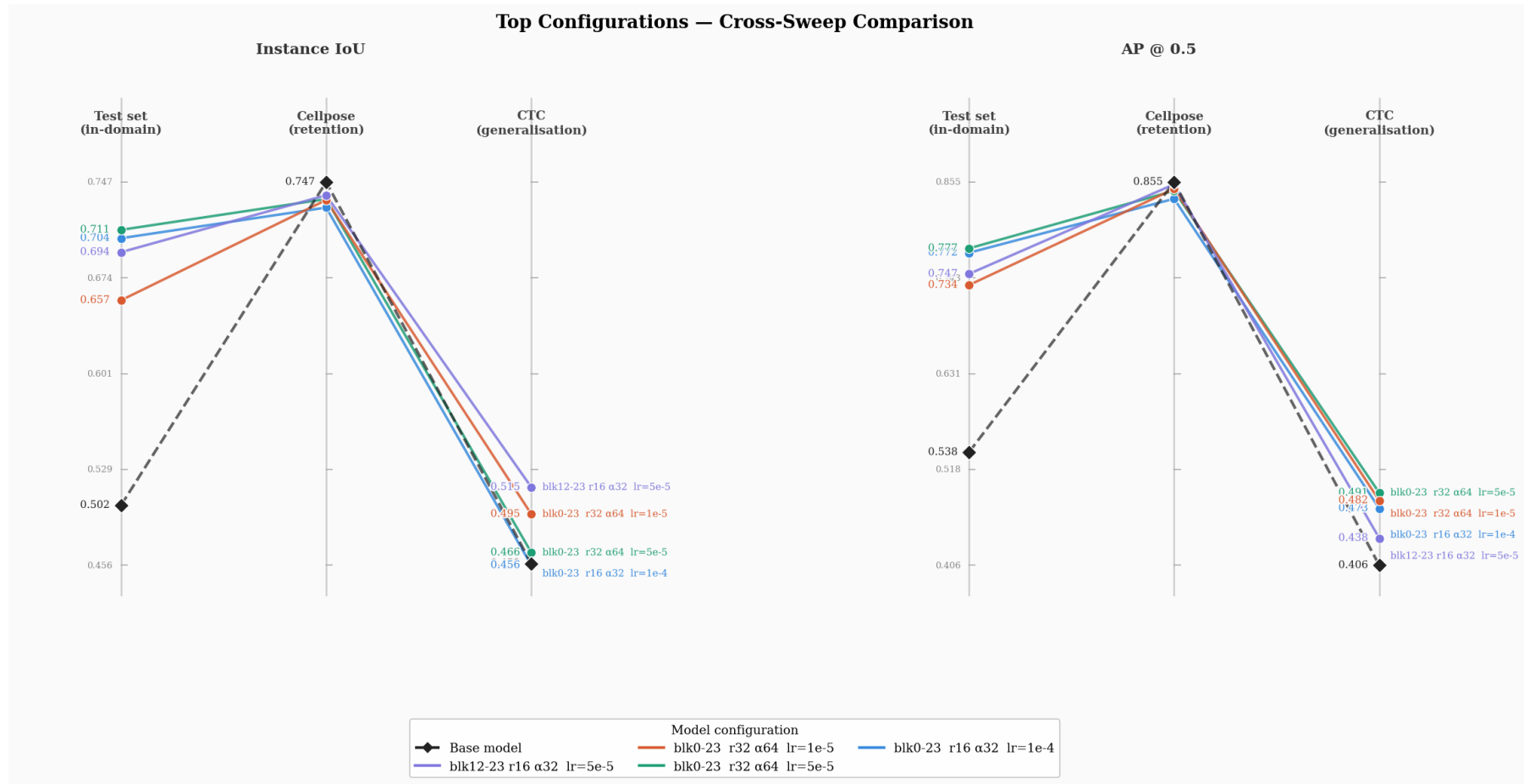


Figure C.4: Comparison of performance when combining the best-performing configurations from previous sweeps.

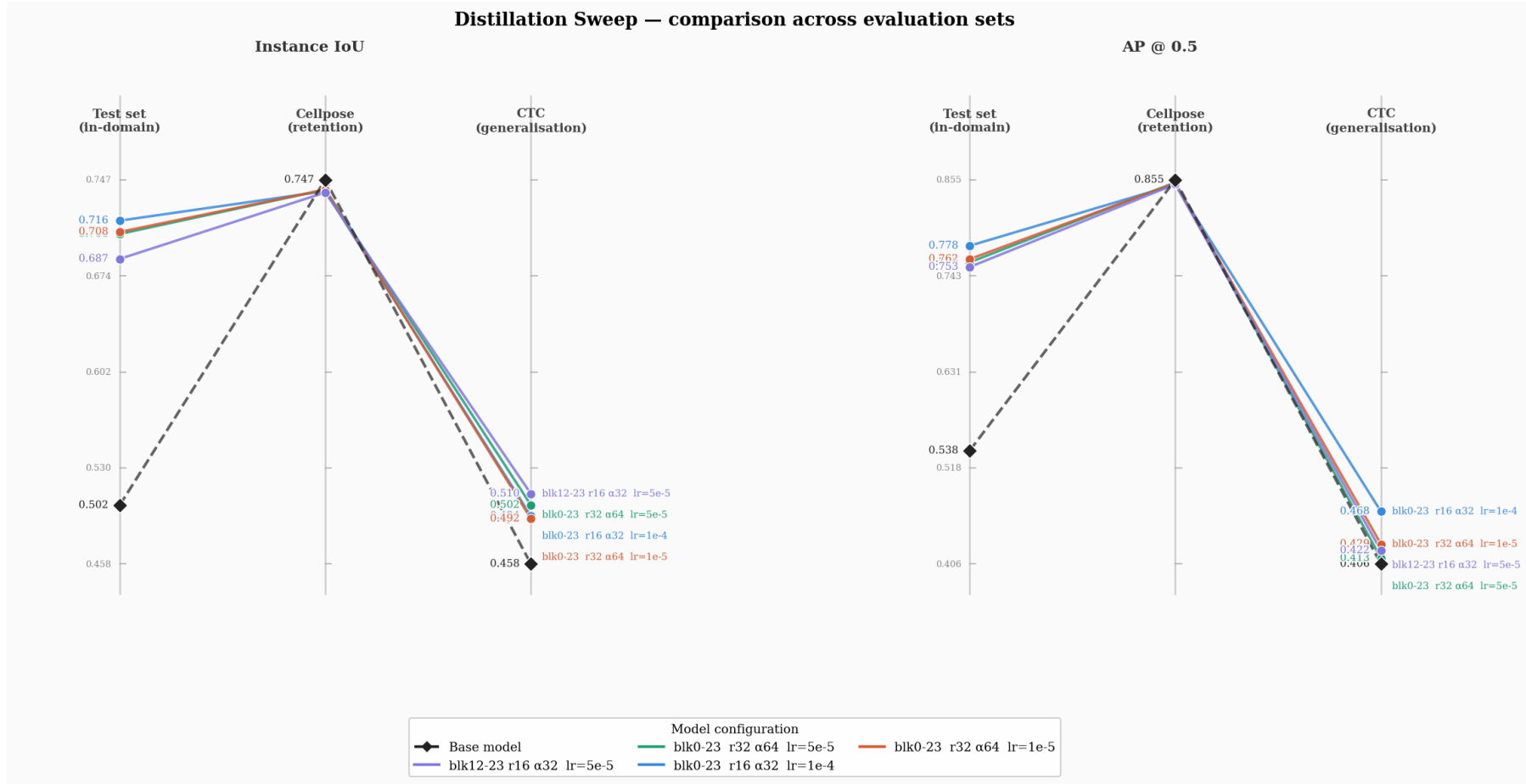


Figure C.5: Comparison of performance utilising feature distillation with configurations mirroring those in the top configuration sweep C.4.

D

Further evaluation results

Table D.1: Comparison of performance between base-model (Base), and final fine-tuned model (FT) on the ALFI test set containing 80 DIC images with GT masks containing Nuclei.

Metric	Base	FT	Δ
Inst IoU	0.1871	0.2203	+0.0332
AP@0.5	0.1323	0.0874	-0.0449
AP@0.75	0.0922	0.0561	-0.0361
AP@0.9	0.0278	0.0176	-0.0102
F1-Error	0.7929	0.8510	+0.0582
Error Rate	1.4051	2.1116	+0.7065

Table D.2: Comparison of performance between base-model (Base), and final fine-tuned model (FT) on the Dataset Phase400 from CellSAM containing 54 PHC images.

Metric	Base	FT	Δ
Inst IoU	0.7695	0.7719	+0.0024
AP@0.5	0.8807	0.8866	+0.0059
AP@0.75	0.6843	0.6483	-0.0361
AP@0.9	0.0742	0.0583	-0.0159
F1-Error	0.0646	0.0609	-0.0037
Error Rate	0.1287	0.1225	-0.0062

Table D.3: Comparison of performance between base-model (Base), and final fine-tuned model (FT) on the StarDistBF dataset containing 27 BF images.

Metric	Base	FT	Δ
Inst IoU	0.5531	0.5631	+0.0100
AP@0.5	0.1331	0.1214	-0.0118
AP@0.75	0.0601	0.0489	-0.0112
AP@0.9	0.0049	0.0034	-0.0014
F1-Error	0.7725	0.7872	+0.0147
Error Rate	6.6700	6.9012	+0.2312

E

Qualitative analysis

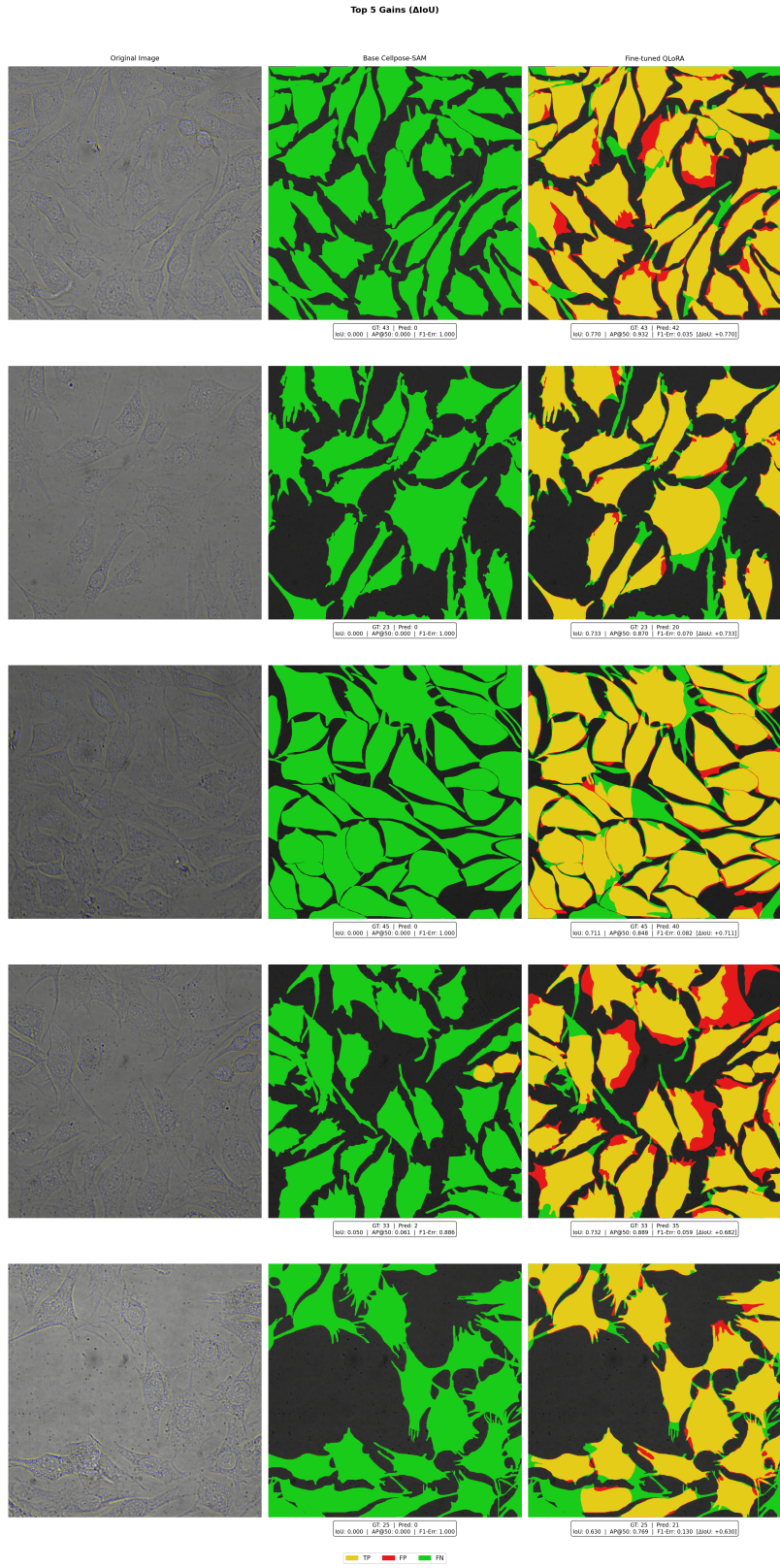


Figure E.1: The 5 images from the fine-tuning test-set images with the highest iIoU-gain

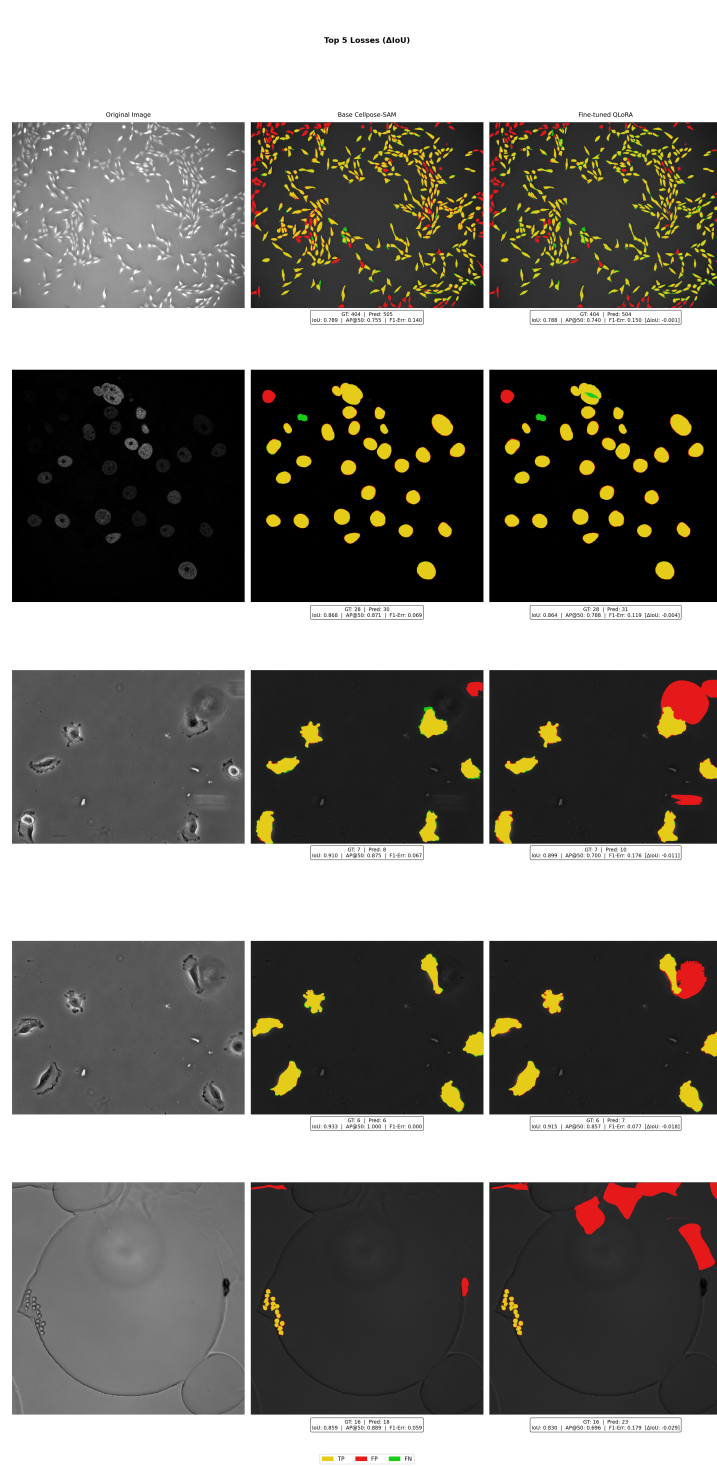


Figure E.2: The 5 images from the fine-tuning test-set with the lowest iIoU-gain

E.1 ALFI

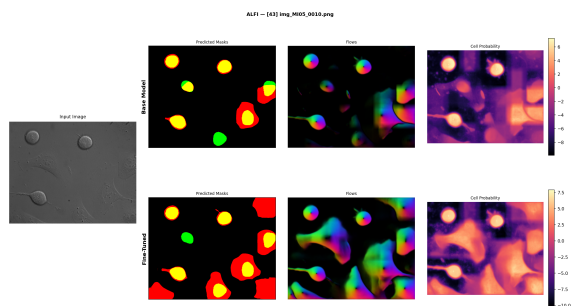


Figure E.3: Segmentation mask, gradient flows and cell probability map for ALFI sample

E.2 Phase400 CellSAM

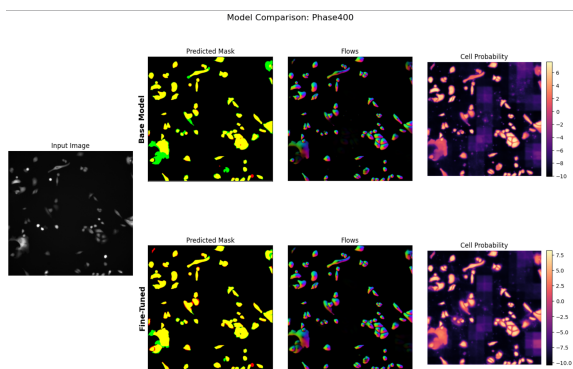


Figure E.4: Segmentation mask, gradient flows and cell probability map for sample from Phase400 dataset