LivecellX: A Deep-learning-based, Single-Cell Object-Oriented Framework for Quantitative Analysis in Live-Cell Imaging

Ke Ni[1,2], Gaohan Yu[3], Zhiqian Zheng[3], Yong Lu[2], Dante Poe[1,2], Shiman Zhou[2], Weikang Wang[4,5*†], Jianhua Xing[2,3,6*†]

[1*]Joint CMU-Pitt Ph.D. Program in Computational Biology, University of Pittsburgh, Pittsburgh, 15232, PA, USA.

[2]Department of Computational and Systems Biology, University of Pittsburgh, Pittsburgh, 15232, PA, USA.

[3]Department of Physics and Astronomy, University of Pittsburgh, Pittsburgh, 15232, PA, USA.

[4]Institute of Theoretical Physics, Chinese Academy of Sciences, Beijing, 100190, China.

[5]School of Physical Sciences, University of Chinese Academy of Sciences, Beijing, 100049, China.

[6]UPMC-Hillman Cancer Center, University of Pittsburgh, Pittsburgh, 15232, PA, USA.

*Corresponding author(s). E-mail(s): wangwk@itp.ac.cn; xing1@pitt.edu;

†These authors contributed equally to this work.

Abstract

Analyzing single-cell dynamics is crucial for understanding developmental biology, cancer biology, and other complex biological processes. This analysis depends on accurately detecting and tracking individual cells across both spatial and temporal scales, with live-cell imaging serving as a key tool. However, extracting reliable dynamic information from live-cell imaging data remains a significant challenge. The task involves constructing long single-cell trajectories and representing dynamic behaviors through multi-dimensional features. Despite recent advances in deep learning-driven segmentation, pre-trained and fine-tuned models often fail to achieve perfect segmentation in live-cell imaging scenarios. The extended duration of live-cell imaging further amplifies segmentation errors, complicating the maintenance of precise and consistent segmentation.

To address these challenges, we introduce LivecellX, a comprehensive framework for live-cell imaging data analysis. LivecellX provides an integrated solution for segmentation, tracking, and dynamic analysis by adopting a single-cell, object-oriented architecture. This architecture not only enhances segmentation and tracking accuracy but also simplifies the extraction of trajectory dynamics, making it easier for users to analyze complex biological processes. Central to LivecellX is the Correct Segmentation Network (CSN), a context-aware, multi-scale machine learning architecture designed to correct segmentation inaccuracies. To effectively apply CSN to large datasets, we developed trajectory-level algorithms that systematically address specific segmentation issues.

To ensure robustness and user accessibility, we developed an asynchronous graphical user interface (GUI) based on Napari, allowing seamless interaction with the data both interactively and programmatically at any stage of the analysis. By combining automated methods with interactive correction capabilities, LivecellX provides a comprehensive solution for precise, large-scale live-cell imaging analysis, empowering researchers to obtain more accurate biological insights.

Keywords: live-cell imaging, correct segmentation network, lineage construction, single cell trajectory

1 Introduction

Live-cell imaging is a fundamental technique for studying the spatial and temporal dynamics of single cells. Compared to fixed imaging and single-cell sequencing, live-cell imaging offers the advantage of capturing the heterogeneity in cellular dynamics [1, 2]. However, there are several major bottlenecks in processing live-cell imaging data. The first is extracting single-cell trajectories from the massive datasets generated by live-cell imaging microscopes. To obtain various types of quantitative information, such as cell morphology and protein distribution across individual cells, accurate single-cell segmentation is crucial, yet it remains one of the most challenging steps [3]. The advent of deep learning, particularly deep convolutional neural networks, has made it feasible to segment extensive single-cell imaging data [4–8]. Nonetheless, no segmentation method can achieve perfect accuracy, especially when training data is limited for label-free livecell datasets and specific cell lines. Analyzing single-cell information necessitates both segmentation and tracking. Segmentation errors, such as over-segmentation (O-Seg) and under-segmentation (U-Seg), are significantly amplified during long-term tracking. For instance, if cells are tracked over 100 frames with a segmentation accuracy of 98%, the probability of maintaining a trajectory without any segmentation errors drops to approximately $0.98^{100} \approx 0.13$. This makes human visual check and manual correction for millions of cells impractical. Therefore, identifying and correcting improperly segmented masks is essential [9]. The challenge is further exacerbated by the cell cycle. Lineage tracing is a critical task in analyzing long-term live-cell imaging data [10], but cell proliferation significantly complicates segmentation and tracking [11]. Firstly, the appearance of cells undergoing division differs markedly from regular cells, complicating the training of deep learning models for segmentation. Secondly, cell division increases cell confluence, further complicating segmentation. Thirdly, cell division adds complexity to tracking. For example, tracking algorithms often rely on a score function based on displacement or overlap between cells in consecutive frames and solve the linear assignment problem to link the cells. The introduction of a new cell instance can create undesired linkages.

The second bottleneck in processing live-cell imaging data is the extraction of high-dimensional features. In fixed imaging and single-cell sequencing, it is possible to extract hundreds of features from snapshot experimental data using labeling. However, extracting high-dimensional features from live-cell imaging data poses significant challenges due to the physical limitations of fluorescence labeling. Firstly, the number of fluorescence labeling channels is limited by the capabilities of fluorescent microscopy and labeling techniques. Secondly, fluorescence labeling can induce phototoxicity in cells during long-term live-cell imaging [10]. In contrast, transmitted light imaging maintains cells in a healthier physical state, significantly enhancing the reliability and repeatability of experiments, particularly for long-term live-cell imaging studies. To overcome the limitations of fluorescence microscopy in live-cell imaging, various methods have been developed. These include deep learning approaches [12, 13] and the extraction of morphological or textural features [14–16] from transmitted light images, enabling the extraction of multi-dimensional features without relying on fluorescence.

Increasingly, researchers recognize the importance of extracting single-cell information from live-cell imaging data. While several software solutions exist for single-cell tracking and trajectory analysis, none fully address the associated challenges simultaneously [11, 17–20]. Most of these tools are designed for cells with stained nuclei [21] and predominantly rely on traditional computation methods, with only a few incorporating deep learning techniques [11, 22].

In this work, we introduce LivecellX, a single-cell object-oriented deep-learning framework designed to tackle these issues. The framework leverages temporal information to identify O-Seg and U-Seg errors and employs a convolutional neural network (CNN) to automatically correct these segmentation errors. Additionally, LivecellX includes modules for mitosis detection, lineage reconstruction, and the extraction of multi-dimensional features such as morphology and texture. We further integrate Napari user interface with data structures implemented in LivecellX with a graphical user interface (GUI).

In summary, LivecellX addresses several complex problems in the analysis of long-term live-cell imaging data. We demonstrate its capabilities through analyses of live-cell imaging data obtained from two types of microscopes: the Nikon Ti2-E, a differential interference contrast microscope, and the Nanolive CX-A, a holotomography microscope.

2 Results

2.1 An Overview of the LivecellX architecture and workflow

In this paper, we introduce a comprehensive workflow for analyzing single-cell trajectories from time-lapse imaging datasets (Fig. 1a). The contribution of LivecellX is threefold: first, we frame a new task specifically designed for live-cell imaging, aimed at resolving U-Seg and O-Seg errors. We develop a deep-learning-based model that effectively addresses these issues, incorporating novel techniques such as the focus mechanism, EDT normalization, and synthetic data sampling. A new benchmark dataset from laboratory is collected from our laboratory and annotated by domain experts for this task. Second, we design the LivecellX architecture, which transitions from an image-matrix-oriented processing paradigm to a single-cell object-oriented paradigm. Finally, we integrate LivecellX with the Napari graphical user interface, allowing users to interactively manipulate data objects.

Following the segmentation stage, LivecellX constructs static single-cell (SC) objects. Its versatile architecture enables each SC object to encapsulate multiple organelle instances within the the object scope. The underlying data structures and SC class methods streamline data analysis in Python notebook environments. To ensure broad compatibility, LivecellX offers a unified interface for loading segmentation results from various sources, including Cellpose, StarDist, and Detectron [8, 23, 24].

To address O-Seg and U-Seg errors, we developed a Correct Segmentation Network (CSN) that has been trained on both publicly available datasets and live single-cell datasets acquired in our laboratory (Fig. 1b). Besides standard segmentation and tracking, we incorporate a corrective module to specifically designed to correct errors in temporal imaging datasets (Fig. 1b). The CSN identifies and rectifies segmentation inaccuracies by analyzing irregular overlap relationships between single-cell masks across consecutive frames (Fig. 1c). We further implements several algorithms to utilize temporal information and CSN to correct trajectory-level errors after the tracking stage.

For the cell tracking stage, we integrated the SORT tracker [25] and developed trajectory-level algorithms to improve tracking quality using the CSN. These algorithms significantly reduce the number of required CSN inferences compared to applying CSN to every single cell. The corrected single-cell trajectories are updated within the LivecellX data structures for downstream analysis.

LivecellX incorporates various feature extraction functions (Fig. 1e) and parallelize the computation process efficiently. For each segmented cell, features are extracted and stored in single-cell objects. Within this module, we implement several widely-used feature extractors, including morphology features and texture features. Furthermore, LivecellX facilitates the application of machine learning methods to extract single cell features. The extracted features are stored in single-cell objects and can be exported to standard file formats such as JSON and CSV.

For dynamics analysis, we created a dedicated module for analyzing single-cell trajectories(Fig. 1f). This module supports dimension reduction methods like principal component analysis (PCA) and UMAP, which project single-cell trajectories into low-dimensional spaces. Researchers can effectively visualize datasets and build downstream analysis models based on the extracted features offered by LivecellX.

2.2 The correct segmentation network revises static label-free image segmentation results

Accurate segmentation is crucial in time-lapse imaging, as even minor errors can significantly impact downstream analyses. For instance, U-Seg can lead to misidentification during cell tracking, resulting in incorrect trajectory data in subsequent frames.

To address this issue, we propose the Correct Segmentation Network (CSN) to rectify O-Seg and U-Seg in single-cell analyses by leveraging temporal context (see Methods and Fig. 2a). The CSN takes both the incorrect single-cell mask and the corresponding region of interest (ROI) in the original images as input, and outputs the corrected mask. Our augmentation converts masks into Euclidean distance maps. Post-processing procedures, such as the watershed algorithm [26], are applied to the probability map output by the model. Additionally, a classifier is trained jointly to categorize the type of segmentation error.

The most common types of segmentation errors are U-Seg and O-Seg (Fig. 2b). We develop a manual annotation program to collect training data by classifying the temporal consistencies. Correct single-cell masks in consecutive trajectory frames typically exhibit a high Intersection over Union (IoU) with one another. In contrast, segmentation errors result in significantly reduced overlap. Leveraging this observation, we propose a pipeline for preparing training data for the CSN. After performing single-cell segmentation or tracking, events with inconsistent IoU values in consecutive frames are identified. Inconsistencies caused by segmentation errors are then selected, and the correct masks are manually labeled.

We identified another type of segmentation error that we denote as O-Seg dropout (Fig. 2c, top). In this error, one part of a cell are segmented, while certain portions of the single cell are missed by the segmentation algorithm. We add O-Seg dropout to our training and testing sample by augmenting O-Seg training samples described in detailed in 5.1. O-Seg dropout samples are essential for correcting both O-Seg and O-Seg dropout cases during inference, especially since there is no direct information indicating which neighboring fragments belong to the same cell in an O-Seg scenario. Without incorporating O-Seg dropout samples, the CSN must rely solely on combining neighboring single-cell masks to identify and rectify O-Seg errors, which is less effective. CSN, capable of correcting O-Seg dropout cases directly, overcome this problem entirely. To enhance data efficiency, we developed a data augmentation method that generates synthetic O-Seg samples based on correctly segmented cells (Fig. 2c, bottom) by dividing them into pieces. Then O-Seg and O-Seg dropout samples were generated from the divided cells.

The CSN data processing pipeline converts both the incorrect input masks and the ground truth masks into Euclidean Distance Transform (EDT) masks [27]. To improve semantic representation, we apply a normalization technique that scales the EDT values to a range of 1 to 5, resulting in contour-map-style inputs and outputs. This normalization helps the CSN focus on pixels requiring correction (Fig. 2d). In contrast, the original EDT has a wide value range, which can complicate the training process and compromise numerical stability.

To improve the Corrective Segmentation Network's (CSN) ability to process multiscale cases, we prepared single-cell image crops at varying scales and trained the model on regions of interest (ROIs) sampled through our augmentation pipeline. This approach enables the CSN to effectively handle inputs of different sizes and spatial resolutions.

We applied the CSN to both differential interference contrast (DIC) images captured with a Nikon Ti-2e microscope and hologram microscope. Here, we present typical examples of corrected O-Seg and U-Seg cases as qualitative evidence of the CSN's effectiveness (Fig. 2e). We evaluated the performance of the CSN using two criteria: 1.Intersection over Union (IoU):This metric represents the average IoU score across all samples, indicating the overlap between ground-truth and predicted cell masks. A higher IoU score signifies better alignment between the two. 2. Cell Match: This criterion measures the percentage of ground-truth cells that are correctly matched to predicted cells, reflecting the accuracy of individual cell identification. Additionally, we assess the percentage of samples in which all ground-truth cells are successfully matched with predicted cells by the CSN. A sample is considered as "solved" if every cell in the ground-truth has a corresponding match in the predicted mask. Our observations indicate that the CSN successfully corrected nearly all the O-Seg cases. For U-Seg, the CSN significantly improved the original masks (Fig. 2f).

Additionally, we have implemented a manual correction module based on Napari[28]. In certain situations, cells can go undetected during segmentation, resulting in the absence of generated masks for these cells. These instances do not fit the categories of O-Seg or U-Seg. To address this, the GUI includes a tool that allows users to draw a coarse mask (polygon) around such undetected cells. The CSN then refines masks based on the user-drawn coarse masks, effectively addressing the issue of undetected cells.

2.3 Trajectory-level correction algorithms accurately identify and correct segmentation errors affecting tracking

Most current cell tracking algorithms are based on object tracking. In the livecell imaging setting, the objects are the cells. If the segmentation mask is wrong in the first place, most of the tracking algorithms cannot rectify the masks. Therefore post-tracking correction is vital.

The typical distortions in linkages between single-cell masks in consecutive frames, caused by O-Seg and U-Seg, are as follows:

1. U-Seg of two cells: When two cells are under-segmented and merged into a single mask, they are mistakenly assigned to the same trajectory, despite belonging to different ones. If the merged object is assigned to trajectory A, a cell in this frame might be incorrectly assigned to trajectory B (Fig. 3a left). This can lead to a cascade of incorrect linkages in other trajectories, particularly those involving neighboring cells.

2. O-Seg of one cell: If a cell at time t in trajectory A is over-segmented into multiple pieces, and only one piece is assigned to trajectory A, the other piece may cause incorrect linkages in the trajectories of neighboring cells (Fig. 3a right). In contrast to U-Seg, O-Seg and O-Seg dropouts may not directly cause incorrect tracking but can affect subsequent analysis of the trajectory.

We design algorithms based on CSN with temporal information to perform tracking corrections (Fig. 3b). By assessing temporal inconsistencies based on the Intersection over Union (IoU) and Intersection over Minimum (IoMin) between consecutive frames of single-cell masks, we can identify candidate segmentation errors. Typically, U-Seg leads to missing frames or fragmented trajectories, whereas O-Seg results in abrupt changes along a trajectory in feature space (Methods). The CSN is applied to these inconsistent cases. Due to consecutive U-Seg cases, we implement a multi-iteration procedure, with results presented in Fig. 3c. The updated trajectories undergo a new cycle of error detection and correction, with iterations continuing until no new errors can be identified.

In Fig. 3c, we present the cumulative counts of U-Seg cases detected and corrected by LivecellX. It should be noted that relying solely on missing or broken trajectories to detect U-Seg cases can lead to numerous false positives. For these false positives, the CSN attempts corrections and generates output masks, which will not be adopted if they conflict with trajectory logic. After fifty iterations, both the attempted corrections and resolved cases reach a point of saturation. For O-Seg, the Receiver Operating Characteristic (ROC) curve indicates that the combination of CSN and trajectory logic achieves a 97% area under curve score.

By incorporating this CSN, we are able to reduce the missing rate (proportion of missing frames) in single cell trajectories (Fig. 3e), which is crucial for understanding the dynamics of cellular behaviors in response to stimuli.

2.4 LivecellX detects biological processes and constructs cell lineages

A key distinction between single-cell time-lapse imaging and other tracking tasks is the presence of the cell cycle. During this process, a mother cell divides into two daughter cells during the mitotic phase. This phase is marked by rapid changes in the cell's appearance, which complicates segmentation and tracking. Nonetheless, mitotic cells in a 2D culture environment exhibit distinct features in transmitted light images (such as those captured by Differential Interference Contrast (DIC) microscopy). Before dividing, the mother cell detaches from the substrate and contracts into a small round shape. Following this, the mother cell divides, creating two daughter cells, each also appearing as a small round shape. Subsequently, the daughter cells begin attaching to the substrate of the Petri dish and spreading out. In transmitted light images, apoptotic cells similarly detach from the substrate and shrink. However, unlike mitotic cells, apoptotic cells have rugged surfaces. This distinction allows apoptotic cells, mitotic cells, and others to be differentiated in transmitted light images More significantly, the birth and termination of trajectories offer distinct dynamic information. Compared to relying solely on static snapshot data, employing temporal information greatly enhances the detection of mitosis and apoptosis (Fig. 4a). To facilitate this, we integrate machine learning based detection methods to identify cells undergoing mitosis, cells in apoptosis, and other cell biological processes (Methods). In essence, this method frames the task of mitosis identification as a trajectory classification problem at the single-cell level based on the dynamics (Fig. 4b). We integrated two types of detection models: One simply relies on the morphological dynamics in cell mitosis and the other one is deep learning based. This process facilitates the construction of cell lineage. Given the critical role of the cell cycle in various biological processes, understanding cell lineage will help elucidate the influence of the cell cycle on these processes. Here, we present a single-cell lineage spanning three generations, captured by livecellX. Representative images at different time points reveal the heterogeneous dynamics of sister cells and cousin cells (Fig. 4c).

2.5 LivecellX enables efficienct trajectory dynamics analysis

By integrating the earlier modules in the package, multiple trajectories can be extracted from time-lapse imaging data. Following segmentation, various properties of individual cell objects can also be retrieved. For transmitted-light images, morphological features hold particular significance. There are various approaches to extracting these features. According to Pincus's report, the active shape model is among the most effective frameworks for representing single-cell morphology [29]. The active shape model, combined with PCA, provides an orthogonal basis set (PCA modes) for describing single-cell morphology (Fig. 1d). Using PCA modes, single-cell morphology can be accurately reconstructed. Furthermore, these modes reflect the dynamic changes in single-cell morphology (cite our own paper). In our implementation, we have incorporated several distinct types of morphological features: the active shape model and predefined features such as area and major axis length (Fig.5a left). In time-lapse imaging experiments, the use of fluorescence-labeled cell lines is common. Previous studies have mainly focused on quantifying fluorescence intensity as the primary measure, which provides limited information. To derive more insights from fluorescence labeling, we have introduced various multi-dimensional features to quantify the distribution of fluorescence proteins across individual cells, based on single-cell segmentation.

These features include Haralick features (Fig. 5a right) and local binary patterns, which are widely used texture features in image analysis. The quantified texture features within a single cell not only encapsulate intensity information but also relate the spatial distribution of fluorescence proteins to cell states. Compared to traditional transmitted-light microscopy, holo-tomography microscopy provides more quantifiable features. For example, organelles, particularly the nucleus, can be clearly imaged. The features of these organelles offer different perspectives for describing cell states. Beyond filter-based features from the traditional computer vision domain, we have also incorporated feature extractors based on machine learning methods such as autoencoders (AE) and variational autoencoders (VAE). The latent embeddings derived from the VAE can serve as features for individual cells. For example, the Harmony method utilizes these embeddings to generate original single-cell images (Fig. 5b, left). Alternatively, a VAE can be employed to generate single-cell masks (Fig. 5b, right), with its latent embeddings capturing different features compared to those obtained through the Harmony method.

With the single-cell feature extractors included in the package, users can analyze single-cell trajectories in high-dimensional feature space. Furthermore, a dynamic manifold can be reconstructed in a topologically equivalent manner using time-delay embedding, as per Takens' theorem [30]. As illustrated in Fig. 5c, the time-delay embedding of a cell cycle trajectory reveals a circle-like structure in the embedded space. The proposed framework allows for statistical analysis of single-cell trajectories. For instance, a histogram plot of cell cycle duration can be generated using LivecellX

(Fig. 5d). To quantify the similarity between multi-dimensional single-cell trajectories, dynamic time warping (DTW) analysis is also included in this pipeline (Fig. 5e, left). DTW can compute the optimal match between trajectories occurring at different speeds. The DTW distance matrix illustrates the heterogeneity of single-cell trajectories throughout the cell cycle (Fig. 5e, middle). Common characteristics of these heterogeneous single-cell trajectories can be revealed through barycenter averaging based on DTW distances (Fig. 5e, right).

2.6 LivecellX data structure encapsulates I/O operations and reduce repetitive temporal data retrieval

In LivecellX, we implemented an innovative data structure for analyzing single-cell dynamics in live cell imaging data. Traditional analysis of live-cell imaging data usually follows the image-oriented structure which involves loading matrices from disk and using multiple nested loops to examine data for cells at each time frame, resulting in redundant and cluttered code. This issue becomes even more pronounced when analyzing data with multiple channels and organelles (Fig. 6a). Instead of directly manipulating matrices, we designed a single-cell object-oriented data structure in LivecellX. It constructs cell objects post-segmentation, creates cell trajectories post-tracking, and organizes trajectories into a collection that enables group operations on single-cell trajectories. As illustrated in Fig. 6b, the data structure comprises a three-tier hierarchy. The first level is the single-cell object, which encompasses information about an individual cell, including the cell ID, time frame, meta information, trajectory label, and extracted features. The second level is the single-cell trajectory. This class contains all the single-cell objects within a single trajectory, along with the trajectory's metadata. Notably, it also includes lineage information, recording the labels of the mother cell trajectory, daughter cell trajectories, and sister cell trajectories. The third level is the single-cell trajectory collection, which includes details about the experimental settings and statistics for all trajectories.

For improved usability, LivecellX includes a graphical user interface (GUI). As shown in Fig. 6c, the GUI features a window for visualizing the original images and provides multiple interactive buttons for performing functions such as mask correction, tracking correction, and manual lineage tracing. If a user finds that a particular single-cell trajectory is inaccurate, the GUI serves as a final verification tool to ensure the quality of the extracted information. For instance, we list several types of manipulations on single cell trajectories in Fig. 6d.

LivecellX abstracts input and output handling through a dynamic dataset class, enabling on-demand retrieval of large-scale imaging data. To optimize efficiency, the dataset class implements a least recently used (LRU) cache, reducing redundant disk access when processing the same or adjacent time-frame images.

Another significant challenge is computational complexity issues. Analyzing thousands of cells on a single core can take hours to days. To alleviate this, LivecellX implements modular and extensible operations that significantly speed up the analysis process. By parallelizing feature computation across multiple cores and computing nodes, we achieve a linear speedup. LivecellX facilitates large-scale analysis of cells from multiple experiments and sources.

While the LivecellX architecture streamlines analysis processes, it cannot completely eliminate the burden of tedious tasks. For annotating datasets, visually checking computation results, and manually interacting with data structures, we developed an asynchronous user interface based on Napari. This GUI can be opened and closed programmatically at any time without interrupting the execution of Python notebooks.

3 Discussion

In this study, we present a comprehensive pipeline for analyzing single-cell trajectories from time-lapse imaging data. We address two major challenges often faced in this field: incorrect segmentation and the identification of mitotic events. By leveraging advancements in deep learning, we have developed specialized neural networks to effectively tackle these issues. In addition to a convolutional neural network (CNN) employed for segmenting transmitted light images, we have trained two additional networks. The first is the CSN, designed to rectify O-Seg and U-Seg errors, thereby improving the quality of single-cell trajectories. CSN operates at the single-cell level, which requires smaller networks and less GPU memory compared to whole-image models. Additionally, it classifies errors and generates pixel-level probability maps, enhancing interpretability. The second network is an integrated CNN that utilizes both temporal and spatial information, or video data, to identify mitotic cells. The incorporation of the CSN and the integrated CNN significantly enhances both the accuracy and reliability of the extracted single-cell trajectories. This pipeline promises substantial benefits for research into the dynamics of single cells.

LivecellX offers a single-cell-object-oriented framework for segmenting cells, tracking their movement, extracting cellular features, and analyzing their trajectories. The goal is to liberate researchers from the tedious technical details and error-prone coding structures often associated with these tasks. However, it's important to acknowledge that algorithms cannot solve every challenge encountered in the analysis of time-lapse imaging data. There are specific situations where manual intervention is necessary. To facilitate this, we have developed a graphical user interface (GUI) that enables researchers to manually correct segmentation errors or other issues that the algorithm cannot resolve. For instance, if a segmentation error results in incorrect linkages between trajectories, researchers can use the GUI for manual corrections. In this study, the use of three neural networks was crucial for achieving accurate results. However, supervised learning approaches depend heavily on training data, and labeling this data can be labor-intensive. To mitigate this burden, we have adopted several strategies to minimize manual labeling efforts. We leveraged publicly available cell image datasets to pretrain the segmentation neural network. Additionally, we developed semi-automatic algorithms to assist in labeling incorrect segmentations as well as mitosis and apoptosis events.

Time-lapse imaging is an essential technique for studying cell dynamics, as it captures information not observable in snapshot data. However, challenges associated with fluorescence labeling and imaging can limit the extraction of comprehensive information from time-lapse data. Our pipeline addresses this by implementing modules to extract multi-dimensional features from both transmitted light and fluorescence images. We acknowledge, though, that this still represents a limited portrayal of the high-dimensional dynamics of cells. Fortunately, advances in deep learning have opened new possibilities for extracting additional information from transmitted light images. For example, some research groups have successfully trained neural networks to identify organelles from these images. In future iterations of this pipeline, we aim to incorporate such features to enhance its functionality further. Overall, our pipeline offers a robust framework for analyzing single-cell trajectories in time-lapse imaging data, effectively addressing challenges in segmentation, mitosis identification, and feature extraction. While manual intervention may be necessary in certain situations, we have devised strategies to minimize the need for extensive manual labeling. With ongoing advancements in deep learning and the integration of additional features, we anticipate further improvements and applications for this pipeline in the future.

4 Figures

5 Methods

5.1 Correct segmentation network

The CSN module operates during inference as follows: After segmenting each frame of images, the original segmentation masks for the whole images are generated. We calculate the overlap relationships between consecutive frames. If a cell mask M_i,t in frame t overlaps with two cell masks M_j,t+1 and $M_k, t+1$ in frame $t+1$, it suggests the possibility of O-Seg in frame $t+1$ or U-Seg in frame t. Similarly, if a cell mask M_i, $t+1$ in frame t+1 overlaps with two cell masks M_j,t and M_k,t in frame t, it indicates potential O-Seg in frame (t) or U-Seg in frame (t+1).

The ROI, encompassing the potentially incorrect segmentation mask and its neighboring area in the corresponding frame, is cropped from the original image and its PM. Unlike typical segmentation tasks, the goal of CSN is to achieve correct segmentation masks based on the PMs. To incorporate the necessary information, we adopt a fusion approach: for the PM within the ROI, pixels of the potential incorrect mask are set to 1, while other pixels are set to -1. The ROI of the raw image and PM are multiplied and used as input for CSN.

Since the pre-trained weights of CSN were obtained from computer vision datasets, the model expects input data structured with three channels. Thus, the input tensor shapes are defined as 3 x H x W, where H and W are the height and width dimensions, respectively. The first channel in the inputs represents the fused version of the raw images, serving as the primary source of information for our baseline model. The other two channels are duplications of the first channel to complete the three-channel structure.

CSN produces three distinct masks as the outcome of our predictions: a segmentation mask, an O-Seg mask, and an U-Seg mask. The segmentation mask represents the likelihood of each pixel belonging to the segmentation class, providing a continuous probability distribution across the image. The U-Seg mask finds regions where segmentation boundaries exceed actual object boundaries. Conversely, the O-Seg mask

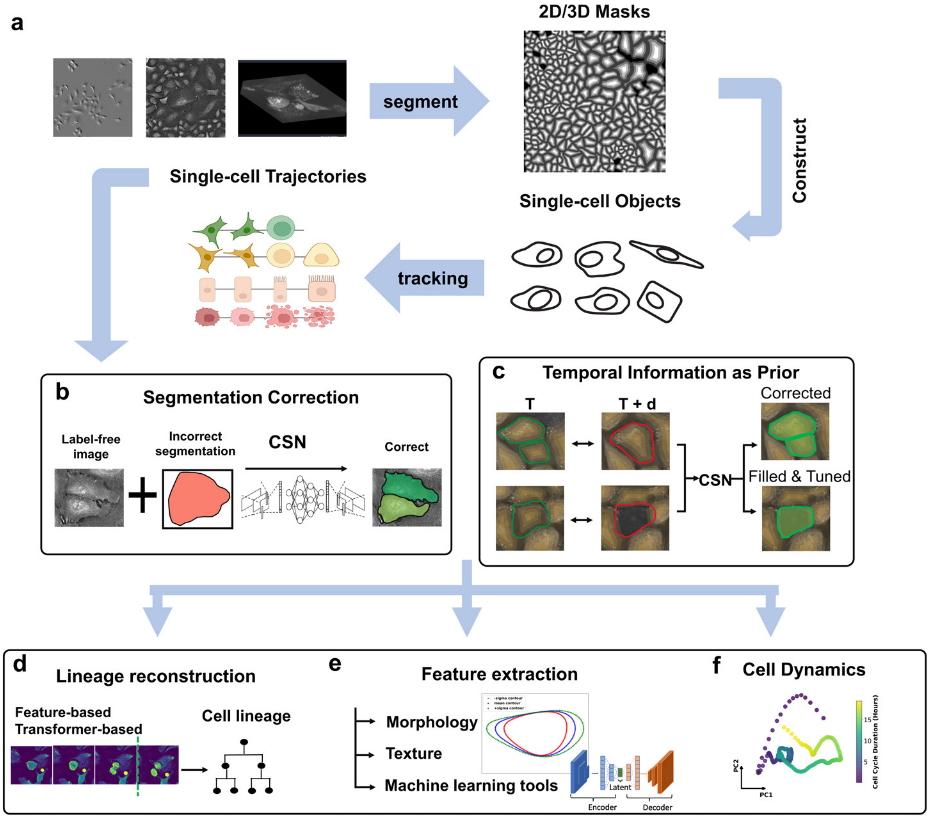


Fig. 1 Overview of the livecellX pipeline for analyzing single-cell dynamics in live-cell imaging data. (a) This comprehensive pipeline encompasses single-cell segmentation, tracking, detection of biological processes, and trajectory analysis. Instead of directly working with matrix data, the LivecellX framework utilizes specifically designed single-cell and trajectory class objects for enhanced analysis. (b) CSN addresses errors in the segmented masks, producing O-Seg and U-Seg probability maps to improve interpretability. (c) LivecellX identifies inconsistencies between time frames and leverages the CSN to correct potential tracking errors. (d) By integrating a deep-learning approach aimed at detecting mitosis and apoptosis—the method labels single-cell trajectories as mother and daughter trajectories based on mitosis detection, facilitating downstream analyses. (e) The pipeline supports various types of single-cell feature extractors, enriching the analysis. (f) Single-cell trajectories are projected into a multi-dimensional feature space for subsequent, in-depth analysis.

identifies regions where the model fails to capture fine details or boundaries within objects. In short, the U-Seg mask is a probability map of the pixels that should be removed. In contrast, the O-Seg mask is a probability map of including pixels.

By generating these three distinct masks, our method enhances the interpretability of segmentation results. The probability mask offers insights into the confidence of model predictions, while the O-Seg and U-Seg masks provide information about

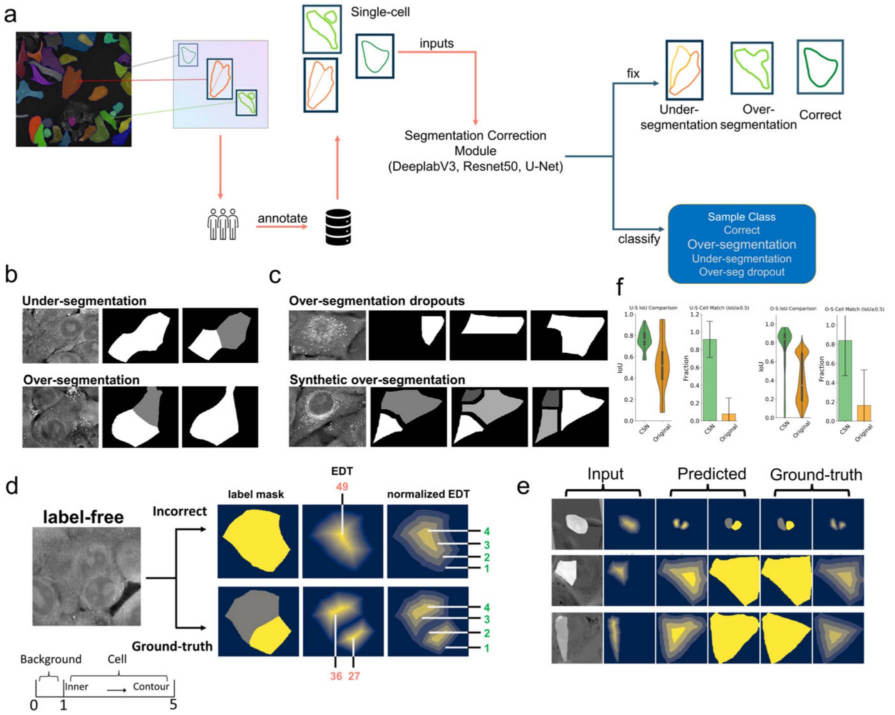


Fig. 2 Architecture of Correct Segmentation Network(CSN). (a) The architecture of the Correct Segmentation Network (CSN), describing its structure, input, and output. The CSN module takes the segmented mask from the initial step and the cropped region of interest (ROI) from label-free imaging results as inputs. (b-c) Our training data includes both real and synthetic datasets. To enhance the practicality of the CSN, we augment the O-Seg samples and introduce O-Seg dropouts, which require the CSN to fill in missing mask regions, making the correction task more challenging. (d) Example of a real training sample demonstrating the superior numerical and semantic stability of our normalized EDT masks compared to the unnormalized version. An incorrect mask (top) is provided as input, and the CSN is tasked with producing the ground truth (bottom). The normalized and unnormalized input-output variants are shown, with min/max values displayed for comparison. (e) Examples of typical outputs from the CSN, showing successful corrections of O-Seg and U-Seg errors. (f) Statistical analysis of the CSN's accuracy, focusing on the sample-level metrics we designed to measure the correction rates for O-Seg and U-Seg.

the model's performance in identifying and resolving subtasks. Together, these masks offer a detailed understanding of our approach's strengths and limitations, facilitating further evaluation and refinement of our segmentation methodology.

5.1.1 Dataset annotationn

We annotate the dataset and generate training samples. Our datasets include instances of both O-Seg and U-Seg. Essentially, we map single objects from the original segmentation masks to corresponding objects in the ground truth, as defined by human

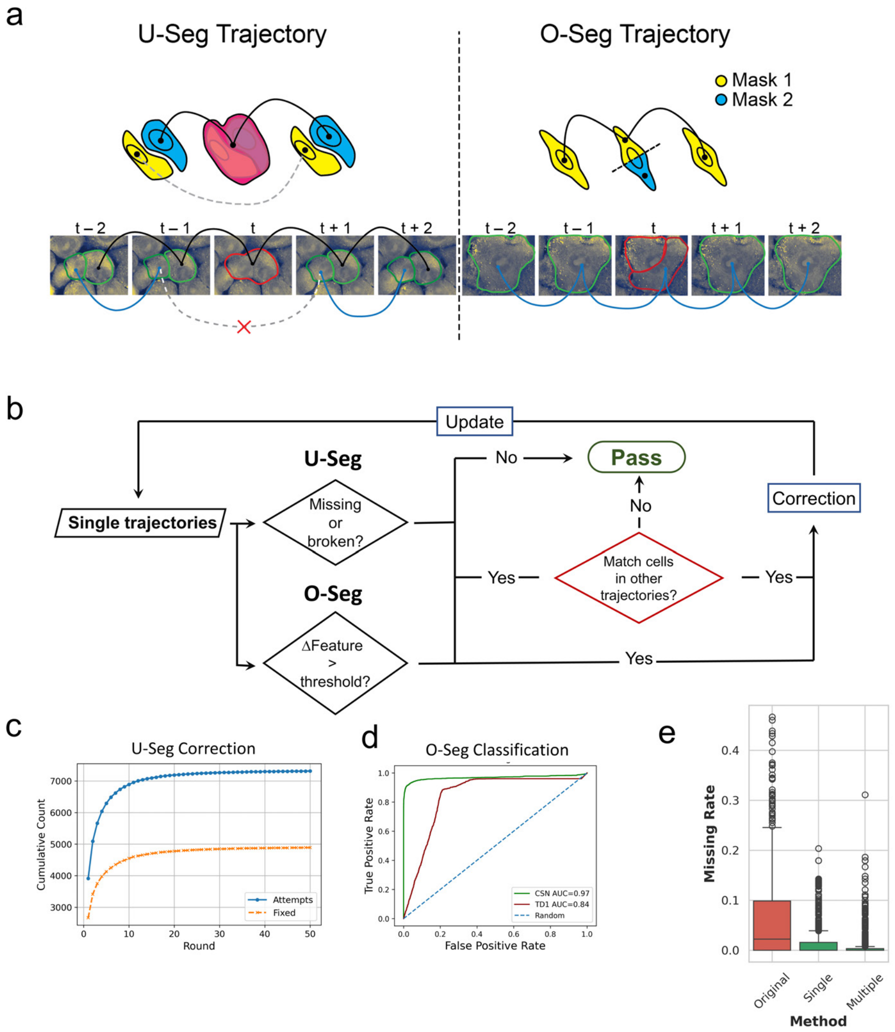


Fig. 3 Figure 3 Temporal Correction on Trajectories (a) Schematics of two idealized O-Seg and U-Seg cases, with real examples (bottom) for illustration: A U-Seg case typically involves two trajectories, resulting in a missing cell time point in one cell trajectory and an extra cell in the other trajectory at the same time point. In contrast, O-Seg usually creates a brief, single-time-point trajectory and leads to inconsistent morphological changes within one trajectory. (b) High-level workflow of trajectory correction logic: For U-Seg cases, our algorithm first identifies candidate trajectories that match the scenarios depicted in (a), specifically those with missing time points or abrupt endings. If the cell mask at the end or missing time point matches a cell mask from another trajectory, the pair is selected as a correction candidate. For O-Seg cases, we filter single-cell trajectories based on feature changes. We then apply CSN to correct the selected candidates and update the single-cell trajectory objects accordingly. (c) Cumulative counts of fix attempts and successfully fixed cases in each round. (d) ROC curve for classifying O-Seg cases based on CSN correction results and a simple two-way feature change score, as defined in the Methods. (e) Statistical results showing that our trajectory-level algorithms significantly reduce the missing rates of trajectory objects. “Single” refers to the single-round correction algorithm variant, while “Multiple” refers to the multi-round algorithm variant.

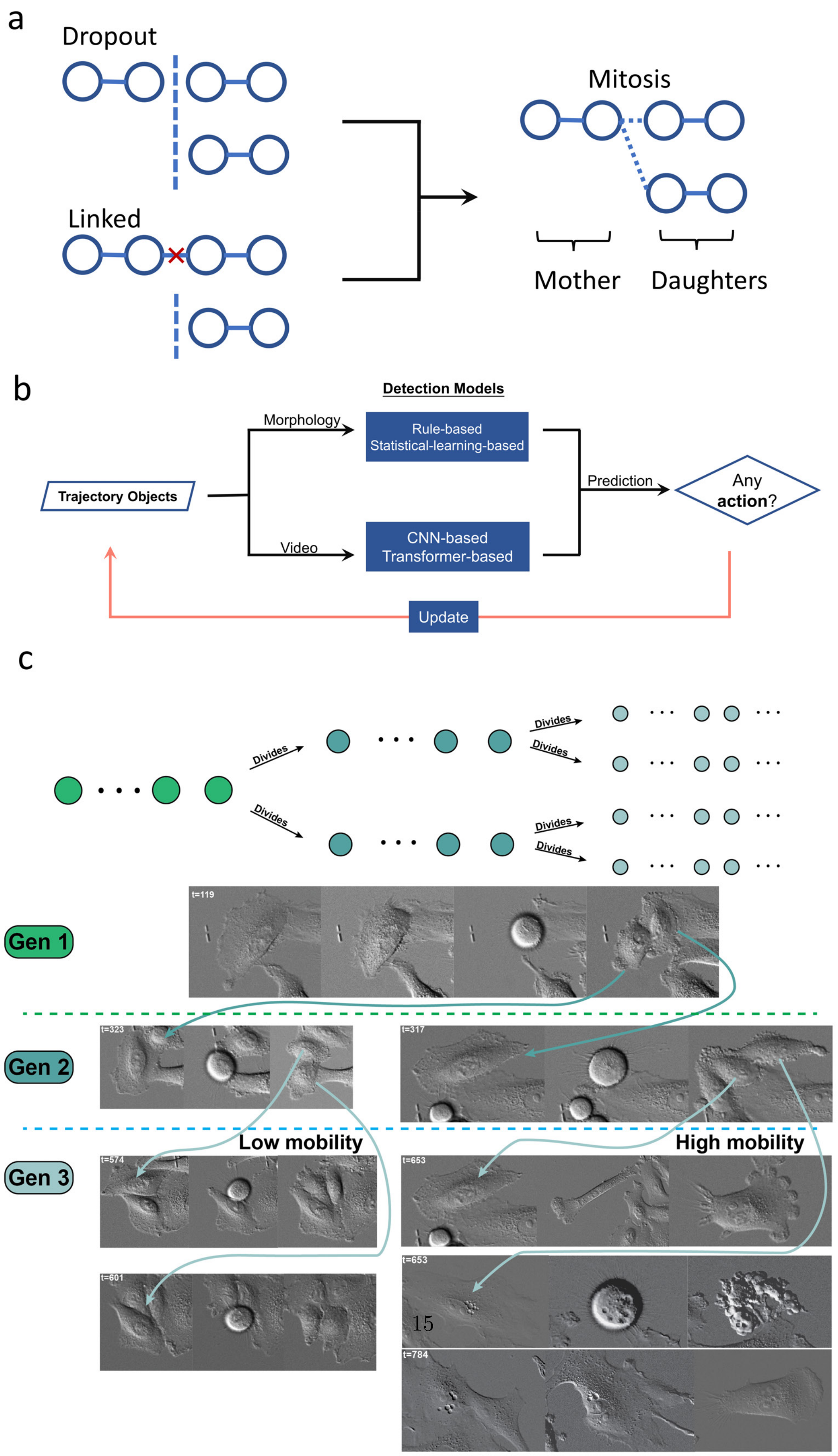


Fig. 4 Lineage reconstruction (a) Typical linkage issues caused by mitosis events. LivecellX detects mitotic cells, splits the detected trajectories and adds lineage information to single cell trajectory objects. (b) Workflow of machine-learning based detection module integrated with LivecellX data structures. (c) Typical examples of a cell lineage, with mitosis-related time frames displayed. Half of the third generation (bottom right) exhibits increased mobility.

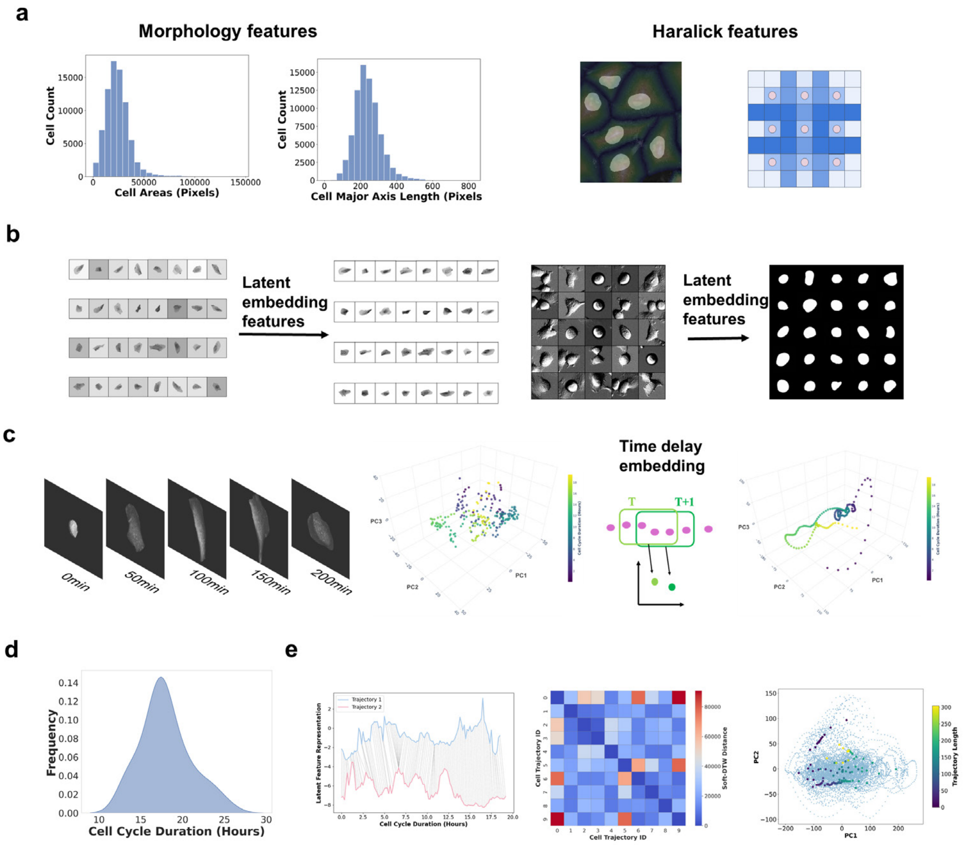
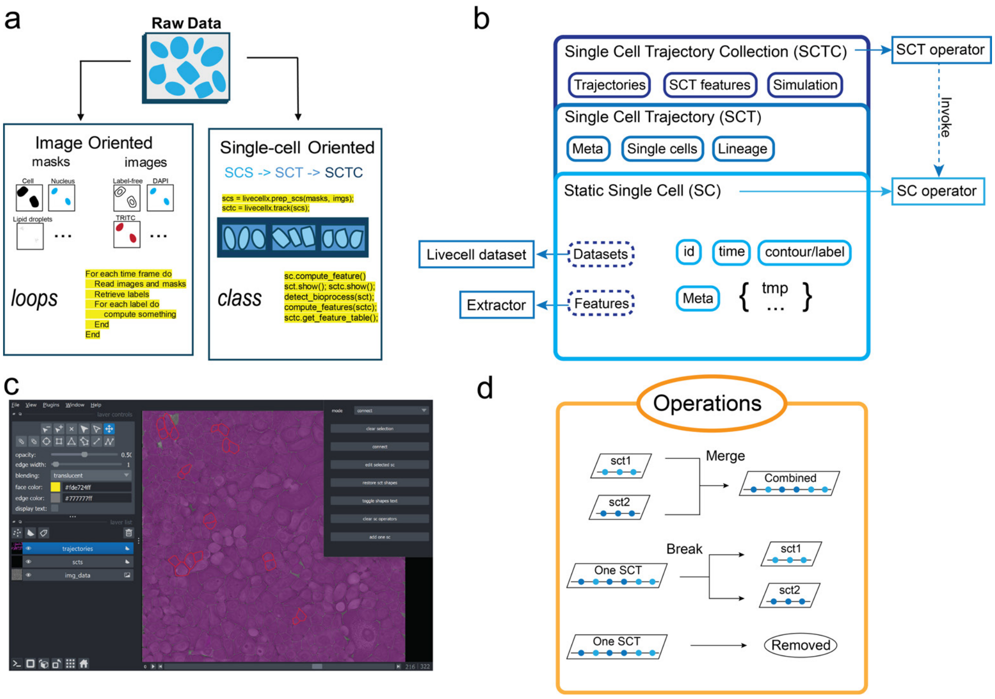


Fig. 5 (a) Mutliple types features can be extracted and saved in single cell object. (b) Feature extraction with latent embedding features of VAE. (c) Time delay embedding of multiple cell cycle trajectories. (d) Distribution of cell cycle duration of A549 cells. (f) Analyzing ensemble of single cell trajectories with DTW. Left: illustration of DTW; middle: DTW distance matrix of mutiple single cell trajectories. Right: averaged dynamics of time-delay cell cycle trajectories.

experts.If one cell object maps onto multiple ground-truth objects, this is defined as an U-Seg sample. Conversely, if multiple cell objects maps to a single ground-truth cell object, this constitutes an O-Seg sample.

5.1.2 Focus mechanism

We introduce a focus mechanism to assign semantic meaning to the tasks. This mechanism is implemented by adding a dedicated channel that aids the deep learning models



```
For each time frame do
   Read images and masks
   Retrieve labels
   For each label do
      compute something
   End
End
```

```
scs = livecellx.prep_scs(masks, imgs);
sctc = livecellx.track(scs);
```

```
sc.compute_feature()
sct.show(); sctc.show();
detect_bioprocess(sct);
compute_features(sctc);
sctc.get_feature_table();
```

opacity: 0.50
edge width: 1
blending: translucent
face color: #fde724ff
edge color: #777777ff
display text:
mode: connect

0 216 | 322

Fig. 6 LivecellX architecture and Napari user interface integration (a) Comparison between image-oriented data structures and single-cell object-oriented data structures. (b) Data structures for static single cells, single-cell trajectories, and single-cell trajectory collections as the foundation of LivecellX. (c) LivecellX integrates the Napari graphical user interface with its underlying data structures. The screenshot shows an interface example, invoked within a Python Jupyter Notebook, for reviewing CSN under-segmentation results. (d) Trajectory operations available in LivecellX, which can be executed either in a Python Jupyter notebook or via the Napari user interface.

in focusing on desired cell objects and organelles. Formally, given an input image tensor $X \in \mathbb{R}^{H\times W\times C}$ and a focus mask $F \in \mathbb{R}^{H\times W\times 1}$, the focused input $\tilde{X}$ is obtained by:

$$\tilde{X} = \text{concat}(X, F),$$

where $\text{concat}(\cdot)$ denotes channel-wise concatenation. The resulting tensor $\tilde{X} \in \mathbb{R}^{H\times W\times (C+1)}$ includes the original image channels along with the focus mask, allowing the model to leverage both visual features and semantic guidance simultaneously during training and inference.

5.1.3 Training and testing

Loss functions

For model training, we employ Binary Cross Entropy (BCE) as our loss function, tailored to meet our specific requirements. Since each pixel may potentially belong to multiple channels, standard cross entropy across all classes is unsuitable. In line with recent advancements in the field [6], we also explore the potential of distance-transformed maps to enhance the network's performance. Accordingly, we compute the Mean Squared Error (MSE) alongside BCE, incorporating Euclidean distance transformation for comparative evaluation.

During our ablation tests, we explore various strategies for augmenting the other two channels. For example, we tested removing the backgrounds from raw images to obtain background-free representations of single cells in one of the duplicated channels. Additionally, we investigated the effectiveness of incorporating segmentation masks and single-cell masks as two additional channels. This approach allows us to evaluate the impact of these extra channels on the model's performance.

5.2 Experimental data collection and livecell imaging setup

Cell Culture A549/VIM-RFP cells (ATCC CCL-185EMT) were cultured with F-12K medium (Corning) with 10% fetal bovine serum (FBS) Cells in glass bottom culture dishes in a humidified atmosphere at 37 °C and 5% CO2. Cells within 3–10 generations were used in this study.

Live-cell cell imaging For live cell imaging with Nikon Ti2-E microscope, 5,000-30,000 cells were seeded into 35 mm Glass bottom dishes (Cellvis D35-20-1.5-N) two days before imaging. Differential interference contrast (DIC) images and TRITC channel (Ex 555 nm/Em 587nm) images were taken with 20× objective(N.A. = 0.75). For the DIC channel, images were taken every five minutes; and for the TRITC channel, images were taken every 30 minutes. The cells were imaged for 24-120 hours. For imaging longer than 48 hours, 500umL culture medium were added every 48 hours.

For live cell imaging with Nanolive-CXA microscope, 500-5,000 cells were seeded into 35 mm imaging dish with a glass bottom (Ibidi, 80137) two days before imaging. Nanolive images and TRITC channel (Ex 555 nm/Em 587nm) images were taken with 60× objective(N.A. = 1.45). For the Nanolive channel, images were taken every five minutes; and for the TRITC channel, images were taken every 30 minutes. The cells were imaged for 24-72 hours. For imaging longer than 48 hours, 300umL culture medium were added every 48 hours.

5.3 Trajectory correction logic

When the majority of cells are correctly segmented, it is unnecessary to apply the CSN to verify and correct every segmented cell, which could involve processing millions of single-cell (SC) objects. In this context, we leverage spatial-temporal information from live cell tracking to refine single-cell trajectories, a methodology we refer to as "trajectory logic". O-Seg typically results in abrupt fluctuations in IoU along single-cell trajectories, often accompanied by an additional mask at the current frame or a short trajectory consisting of a partial mask that persists for several frames. Errors arising from O-Seg are identified and corrected based on this logical framework. Addressing U-Seg poses a greater challenge than managing O-Seg, as the former often results in cascading tracking errors among neighboring trajectories. The primary types of these errors include the omission of specific frames in adjacent single-cell trajectories and the premature termination of trajectories before the experiment concludes. Consequently, under-segmented cell masks can be identified by recognizing these errors. Understanding the nature of these errors is also essential for accurately mapping the corrected masks back into the single-cell trajectories. To filter out missing and early-ending events, we utilize IoU and IoMin metrics between single-cell masks across consecutive frames. The corrected masks are then integrated into single-cell trajectories according to the identified error types. For example, if an under-segmented cell mask at frame t is corrected into two distinct cell masks, and one of these can be mapped to a neighboring single-cell trajectory that is missing a frame at the same t, then the trajectory logic remains self-consistent, increasing the likelihood of accurate trajectories.

If the CSN's results are not aligned with the spatial-temporal information derived from local imaging data, the proposed corrections should not be adopted. Additionally, some errors may persist across multiple frames. To address this, we have adopted an iterative strategy, allowing both the CSN and the trajectory correction logic processes to run through multiple iterations until no further corrections can be made.

5.4 Feature extraction and trajectory dynamics

In this method, we offer several modules for feature extraction, including the active shape model, Haralick features, local binary pattern, variational auto-encoder, Harmony [31], and diffusion models [32, 33]. LivecellX integrates the original active shape model implemented in Pincus's Celltool[29]. Haralick features and local binary patterns are extracted using the Mahotas package[34]. The variational auto-encoder is implemented through Harmony. In the CSN model, regions containing potential error masks, along with padding, are cropped from the original images. This ensures that the ROIs include information about neighboring cells and the background. However, the CSN is tasked with correcting the masks of cells located at the center of these ROIs. The focus channel helps direct the CSN's attention to these central cells. For example, when addressing an U-Seg issue, the normalized Euclidean Distance Transform (EDT) mask of the under-segmented cell is added as the focus channel in the input. Pixels outside the normalized EDT mask are set to zero.

For lineage construction, the trajectory object's input also includes information about neighboring cells and the background. To ensure the Timesformer model focuses on potential mitosis events, the normalized EDT mask of each cell along the trajectory is added as a focus channel.

This additional channel significantly enhances the efficiency of both correction and detection processes.

Acknowledgements. This work was partially supported by National Institute of General Medical Sciences (R01GM148525), National Institute of Diabetes and Digestive and Kidney Diseases (R01DK1192, R56DK119232) and National Science Foundation (DMS2325149) to JX, National Institute of Biomedical Imaging and Bioengineering (T32EB009403) to DP, National Natural Science Foundation of China Grants (No.12247104) to WW.

Declarations

The authors declares no conflict of interest.

References

[1] Gordonov, S., Hwang, M.K., Wells, A., Gertler, F.B., Lauffenburger, D.A., Bathe, M.: Time series modeling of live-cell shape dynamics for image-based phenotypic profiling. Integr Biol (Camb) **8**(1), 73–90 (2016) https://doi.org/10.1039/C5IB00239A

[2] Wang, W., Douglas, D., Zhang, J., Kumari, S., Enuameh, M.S., Dai, Y., al.: Live-cell imaging and analysis reveal cell phenotypic transition dynamics inherently missing in snapshot data. Science Advances **6**(36), 9319 (2020) https://doi.org/10.1126/sciadv.aba9319

[3] Uchida, S.: Image processing and recognition for biological images. Dev Growth Differ **55**(4), 523–549 (2013) https://doi.org/10.1111/dgd.12033

[4] Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(4), 640–651 (2017) https://doi.org/10.1109/TPAMI.2016.2572683

[5] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention – MICCAI, pp. 234–241. Springer, ??? (2015). https://doi.org/10.1007/978-3-319-24574-4_28

[6] Wang, W., Taft, D.A., Chen, Y.-J., Zhang, J., Wallace, C.T., Xu, M., al.: Learn to segment single cells with deep distance estimator and deep cell detector. Computers in Biology and Medicine **108**, 133–141 (2019) https://doi.org/10.1016/j.compbiomed.2019.03.011

[7] Van Valen, D.A., Kudo, T., Lane, K.M., Macklin, D.N., Quach, N.T., DeFelice, M.M., al.: Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments. PLOS Computational Biology **12**(11), 1005177 (2016) https://doi.org/10.1371/journal.pcbi.1005177

[8] Stringer, C., Wang, T., Michaelos, M., Pachitariu, M.: Cellpose: a generalist algorithm for cellular segmentation. Nature Methods **18**, 100–106 (2021) https://doi.org/10.1038/s41592-020-01018-x

[9] Han, H., Wu, G., Li, Y., Zi, Z.: edetect: A fast error detection and correction tool for live cell imaging data analysis. iScience **13**, 1–8 (2019) https://doi.org/10.1016/j.isci.2019.04.005

[10] Skylaki, S., Hilsenbeck, O., Schroeder, T.: Challenges in long-term imaging and quantification of single-cell dynamics. Nature Biotechnology **34**(11), 1137–1144 (2016) https://doi.org/10.1038/nbt.3664

[11] Moen, E., Borba, E., Miller, G., Schwartz, M., Bannon, D., Koe, N., al.: Accurate

cell tracking and lineage construction in live-cell imaging experiments with deep learning. bioRxiv, 803205 (2019) https://doi.org/10.1101/803205

[12] Christiansen, E.M., Yang, S.J., Ando, D.M., Javaherian, A., Skibinski, G., Lipnick, S., al.: In silico labeling: Predicting fluorescent labels in unlabeled images. Cell **173**(3), 792–80319 (2018) https://doi.org/10.1016/j.cell.2018.03.047

[13] Ounkomol, C., Seshamani, S., Maleckar, M.M., Collman, F., Johnson, G.R.: Label-free prediction of three-dimensional fluorescence images from transmitted-light microscopy. Nature Methods **15**(11), 917–920 (2018) https://doi.org/10.1038/s41592-018-0151-5

[14] Bray, M.-A., Singh, S., Han, H., Davis, C.T., Borgeson, B., Hartland, C., al.: Cell painting, a high-content image-based assay for morphological profiling using multiplexed fluorescent dyes. Nature Protocols **11**(9), 1757–1774 (2016) https://doi.org/10.1038/nprot.2016.105

[15] Way, G.P., Natoli, T., Adeboye, A., Litichevskiy, L., Yang, A., Lu, X., al.: Morphology and gene expression profiling provide complementary information for mapping cell state. Cell Systems **13**(11), 911–9239 (2022) https://doi.org/10.1016/j.cels.2022.10.014

[16] Gut, G., Herrmann, M.D., Pelkmans, L.: Multiplexed protein maps link subcellular organization to cellular states. Science **361**(6401), 7173 (2018) https://doi.org/10.1126/science.aar7173

[17] Hu, T., Xu, S., Wei, L., Zhang, X., Wang, X.: Celltracker: an automated toolbox for single-cell segmentation and tracking of time-lapse microscopy images. Bioinformatics **37**(2), 285–287 (2021) https://doi.org/10.1093/bioinformatics/btaa835

[18] Amat, F., Lemon, W., Mossing, D.P., McDole, K., Wan, Y., Branson, K., al.: Fast, accurate reconstruction of cell lineages from large-scale fluorescence microscopy data. Nature Methods **11**(9), 951–958 (2014) https://doi.org/10.1038/nmeth.2985

[19] Padovani, F., Mairhörmann, B., Falter-Braun, P., Lengefeld, J., Schmoller, K.M.: Segmentation, tracking and cell cycle analysis of live-cell imaging data with cell-acdc. BMC Biology **20**(1), 174 (2022) https://doi.org/10.1186/s12915-022-01484-1

[20] Piltti, K.M., Cummings, B.J., Carta, K., Manughian-Peter, A., Worne, C.L., Singh, K., al.: Live-cell time-lapse imaging and single-cell tracking of in vitro cultured neural stem cells - tools for analyzing dynamics of cell cycle, migration, and lineage selection. Methods **133**, 81–90 (2018) https://doi.org/10.1016/j.ymeth.2017.12.003

[21] Tian, C., Yang, C., Spencer, S.L.: Elliptrack: A global-local cell-tracking pipeline for 2d fluorescence time-lapse microscopy. Cell Reports **32**(5), 107984 (2020) https://doi.org/10.1016/j.celrep.2020.107984

[22] Hernandez, D.E., Chen, S.W., Hunter, E.E., Steager, E.B., Kumar, V.: Cell tracking with deep learning and the viterbi algorithm. In: 2018 International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS), pp. 1–7 (2018)

[23] Weigert, M., Schmidt, U., Haase, R., Sugawara, K., Myers, G.: Star-convex polyhedra for 3d object detection and segmentation in microscopy. In: The IEEE Winter Conference on Applications of Computer Vision (WACV) (2020). https://doi.org/10.1109/WACV45572.2020.9093435

[24] Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., Girshick, R.: Detectron2. https://github.com/facebookresearch/detectron2 (2019)

[25] Upcroft, A., Zaga, G., La, O., Fry, B.: Simple online and realtime tracking (2016)

[26] Vincent, L., Soille, P.: Watersheds in digital spaces: An efficient algorithm based on immersion simulations. IEEE Transactions on Pattern Analysis and Machine Intelligence **13**(6), 583–598 (1991)

[27] Felzenszwalb, P.F., Huttenlocher, D.P.: Distance transforms of sampled functions. Theory of Computing **1**(1), 1–48 (2004)

[28] napari contributors: napari: a multi-dimensional image viewer for python. https://doi.org/10.5281/zenodo.3555620 (2019). https://doi.org/10.5281/zenodo.3555620

[29] Pincus, D., Pincus, S.: Comparison of quantitative methods for cell-shape analysis. Journal of Microscopy **227**(2), 140–156 (2007)

[30] Takens, F.: Detecting strange attractors in turbulence. In: Rand, D., Young, L.-S. (eds.) Dynamical Systems and Turbulence, Warwick 1980, pp. 366–381. Springer, Berlin, Heidelberg (1981)

[31] Uddin, M.R., Howe, G., Zeng, X., Xu, M.: Harmony: A generic unsupervised approach for disentangling semantic content from parameterized transformations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 20614–20623. IEEE, ??? (2022). https://doi.org/10.1109/cvpr52688.2022.01999

[32] Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 6840–6851 (2020)

[33] Fick, A.: On liquid diffusion. Annalen der Physik **170**(1), 59–86 (1855)

[34] Coelho, L.P.: Mahotas: Open source software for scriptable computer vision. Journal of Open Research Software **1**(1), 3 (2013) https://doi.org/10.5334/jors.ac