nature machine intelligence

Article

https://doi.org/10.1038/s42256-024-00903-w

Development of AI-assisted microscopy frameworks through realistic simulation with pySTED

Received: 25 March 2024

Accepted: 20 August 2024

Published online: 26 September 2024

Check for updates

Anthony Bilodeau[1,2], Albert Michaud-Gagnon[1,2], Julia Chabbert[1], Benoit Turcotte[1,2], Jörn Heine[3], Audrey Durand[2,4,5,6] & Flavie Lavoie-Cardinal[1,2,7]

The integration of artificial intelligence into microscopy systems significantly enhances performance, optimizing both image acquisition and analysis phases. Development of artificial intelligence-assisted super-resolution microscopy is often limited by access to large biological datasets, as well as by difficulties to benchmark and compare approaches on heterogeneous samples. We demonstrate the benefits of a realistic stimulated emission depletion microscopy simulation platform, pySTED, for the development and deployment of artificial intelligence strategies for super-resolution microscopy. pySTED integrates theoretically and empirically validated models for photobleaching and point spread function generation in stimulated emission depletion microscopy, as well as simulating realistic point-scanning dynamics and using a deep learning model to replicate the underlying structures of real images. This simulation environment can be used for data augmentation to train deep neural networks, for the development of online optimization strategies and to train reinforcement learning models. Using pySTED as a training environment allows the reinforcement learning models to bridge the gap between simulation and reality, as showcased by its successful deployment on a real microscope system without fine tuning.

Super-resolution microscopy has played a pivotal role in life sciences by allowing the investigation of the nano-organization of biological samples to a few tens of nanometres[1]. Stimulated emission depletion (STED)[2], a point scanning-based super-resolution microscopy fluorescence modality, routinely allows resolution down to 30–80 nm to be reached in fixed and live samples[1]. One drawback of STED microscopy is the photobleaching of fluorophores associated with increased light exposure at the sample[1,3,4]. Photobleaching results in a decrease in fluorescence, limiting the ability to capture multiple consecutive images of a particular area and may also increase phototoxicity in living samples[4,5]. In an imaging experiment, photobleaching and phototoxicity need to be minimized by the careful modulation of imaging parameters[5,6] or by adopting smart-scanning schemes[7–9]. Integration of artificial intelligence (AI)-assisted smart modules to bioimaging acquisition protocols has been proposed to guide and control microscopy experiments[6,7,10,11]. However, machine learning (ML) and deep learning (DL) algorithms generally require a large amount of annotated data to be trained, which can be difficult to obtain when working

[1]CERVO Brain Research Center, Québec, Québec, Canada. [2]Institute for Intelligence and Data, Québec, Québec, Canada. [3]Abberior Instruments GmbH, Göttingen, Germany. [4]Department of Computer Science and Software Engineering, Université Laval, Québec, Québec, Canada. [5]Department of Electrical and Computer Engineering, Université Laval, Québec, Québec, Canada. [6]Canada CIFAR AI Chair, Mila, Québec, Canada. [7]Department of Psychiatry and Neuroscience, Université Laval, Québec, Québec, Canada. e-mail: flavie.lavoie-cardinal@cervo.ulaval.ca

with biological samples. Diversity in curated training datasets also enhances the model's robustness[12,13]. Although large annotated datasets of diffraction-limited optical microscopy have been published in recent years[14,15], access to such datasets for super-resolution microscopy is still limited, in part due to the complexity of data acquisition and annotation as well as limited access to imaging resources. Similarly, the development of reinforcement learning (RL) methods adapted to the control of complex systems on a wide variety of tasks in games, robotics or even in microscopy imaging, are strongly dependent on the availability of large training datasets, generally relying on the development of accessible, realistic and modular simulation environments[11,16–20]

To circumvent this limitation, simulation strategies have been used for high-end microscopy techniques. For instance, in fluorescence lifetime imaging microscopy, it is common practice to use simulation software to generate synthetic measurements to train ML/DL models[21]. The models can be completely trained in simulation or with few real measurements. Researchers in single-molecule localization microscopy have also adopted simulation tools in their image analysis pipelines to benchmark their algorithms[22–24]. In an earlier work[25], a DL model was trained with simulated ground-truth detections and few experimental images, which was then deployed on real images. In STED microscopy, simulation software are also available. However, they are limited to theoretical models of the point spread function (PSF)[26,27] or effective PSF (E-PSF)[8,28], without reproducing realistic experimental settings influencing the design of STED acquisitions (for example, photobleaching, structures of interest and scanning schemes). This limits the generation of simulated STED datasets and associated training of ML/DL models for smart STED microscopy modules.

We created a simulation platform, pySTED, that emulates an in silico STED microscope with the aim to assist in the development of AI methods. pySTED is founded on theoretical and empirically validated models that encompass the generation of E-PSF in STED microscopy, as well as a photobleaching model[3,19,26,29]. Additionally, it implements realistic point-scanning dynamics in the simulation process, allowing adaptive scanning schemes and non-uniform photobleaching effects to be mimicked. Realistic samples are simulated in pySTED by using a DL model that predicts the underlying structure (data maps) of real images.

pySTED can benefit the STED and ML communities by facilitating the development and deployment of AI-assisted super-resolution microscopy approaches (Extended Data Fig. 1). It is implemented in a CoLaboratory notebook to help trainees develop their intuition regarding STED microscopy on a simulated system (Extended Data Fig. 1(i)). We demonstrate how the performance of a DL model trained on a semantic segmentation task of nanostructures can be increased using synthetic images from pySTED (Extended Data Fig. 1(ii)). A second experiment shows how our simulation environment can be leveraged to thoroughly validate the development of AI methods and challenge their robustness before deploying them in a real-life scenario (Extended Data Fig. 1(iii)). Last, we show that pySTED enables the training of an RL agent that can learn by interacting with the realistic STED environment, which would not be possible on a real system due to data constraints[30]. The resulting trained agent can be deployed in real experimental conditions to resolve nanostructures and recover biologically relevant features by bridging the reality gap (Extended Data Fig. 1(iv)).

STED simulation with pySTED

We have built a realistic, open-sourced STED simulation platform within the Python[31] environment, namely, pySTED. pySTED breaks down STED acquisition into its main constituents: wavelength-dependent focusing properties of the objective lens, fluorophore excitation and depletion, and fluorescence detection. Each step of the acquisition process corresponds to an independent component of the pipeline and is created with its own parameters (Supplementary Tables 1–4) that users can modify according to their experimental requirements (Fig. 1a)[26].

Generating a synthetic image with the pySTED simulator requires a map of the emitters in the field of view and specify the photophysical properties of the fluorophore (Fig. 1a and Supplementary Table 5). The map of fluorophores, referred to as a data map, can consist of automatically generated simple patterns (for example, beads and fibres) or more complex structures generated from real images (Methods). The emission and photobleaching properties of the fluorophores that are implemented in pySTED are inspired from previous theoretical and experimental models[3,29]. As in a real experiment, the data map is continuously being updated during the simulation process to realistically simulate point-scanning acquisition schemes (Fig. 1a–e and Methods).

Realistic data map generation: data maps that can reproduce diverse biological structures of interest are required for the development of a simulation platform that enables the generation of realistic synthetic STED images. Combining primary object shapes such as points, fibres or polygonal structures is efficient and simple for some use cases, but is not sufficient to represent more complex and diverse structures that can be found in real biological samples[22–24,32]. It is essential to reduce the gap between simulation and reality for microscopist trainees or to train AI models on synthetic samples before deployment on real tasks[33,34].

We sought to generate realistic data maps by training a DL model to predict the underlying structures from real STED images, which can then be used in synthetic pySTED acquisition. We chose the U-Net architecture, U-Net$_{\text{data map}}$, as it has been shown to perform well on various microscopy datasets of limited size[35,36] (Fig. 1f). We adapted a previously established approach in which a low-resolution image is mapped onto a resolution-enhanced image[37,38]. Once convolved with an equivalent optical transfer function, the resolution-enhanced synthetic image is compared with the original image.

Here we trained U-Net$_{\text{data map}}$ on the STED images of proteins in cultured hippocampal neurons (Methods, Supplementary Fig. 1 and Supplementary Table 6). During the training process, the model aims at predicting the underlying structure (data map) such that the convolution of the approximated PSF of the STED microscope (full-width at half-maximum of ~50 nm, as measured from the full-width at half-maximum of real STED images) minimizes the mean quadratic error with the real image (Fig. 1f). After training, given a real image, U-Net$_{\text{data map}}$ generates the underlying structure (Supplementary Fig. 1). From this data map, a synthetic pySTED image can be simulated with different imaging parameters (low or high resolution). Qualitative comparison of the synthetic images acquired in pySTED with the real STED images (Supplementary Fig. 1) shows similar super-resolved structures for different neuronal proteins, confirming the capability of U-Net$_{\text{data map}}$ to predict a realistic data map. We also evaluated the quality of images resulting from data maps generated with U-Net$_{\text{data map}}$ or a conventional Richardson–Lucy deconvolution (Methods and Supplementary Fig. 2a). As highlighted in Supplementary Fig. 2b,c, the use of U-Net$_{\text{data map}}$ instead of Richardson–Lucy deconvolution to generate data maps in pySTED results in improved synthetic images.

For the validation of pySTED with a real STED microscope, we characterized the capacities of pySTED to simulate realistic fluorophore properties by comparing the synthetic pySTED images with real STED microscopy acquisitions (Supplementary Table 7). We acquired STED images of the protein bassoon, which had been immunostained with the fluorophore ATTO-647N. We compared the effect of varying the imaging parameters on the pySTED simulation environment and on the real microscope (Supplementary Figs. 3–5). For pySTED, we used the photophysical properties of the fluorophore ATTO-647N from the literature (Supplementary Table 5)[3,39]. The photobleaching constants (k_1 and b) were estimated from the experimental data by using a least-squares fitting method (Methods). Synthetic data maps were generated with U-Net$_{\text{data map}}$ to facilitate a comparison between simulation and reality.

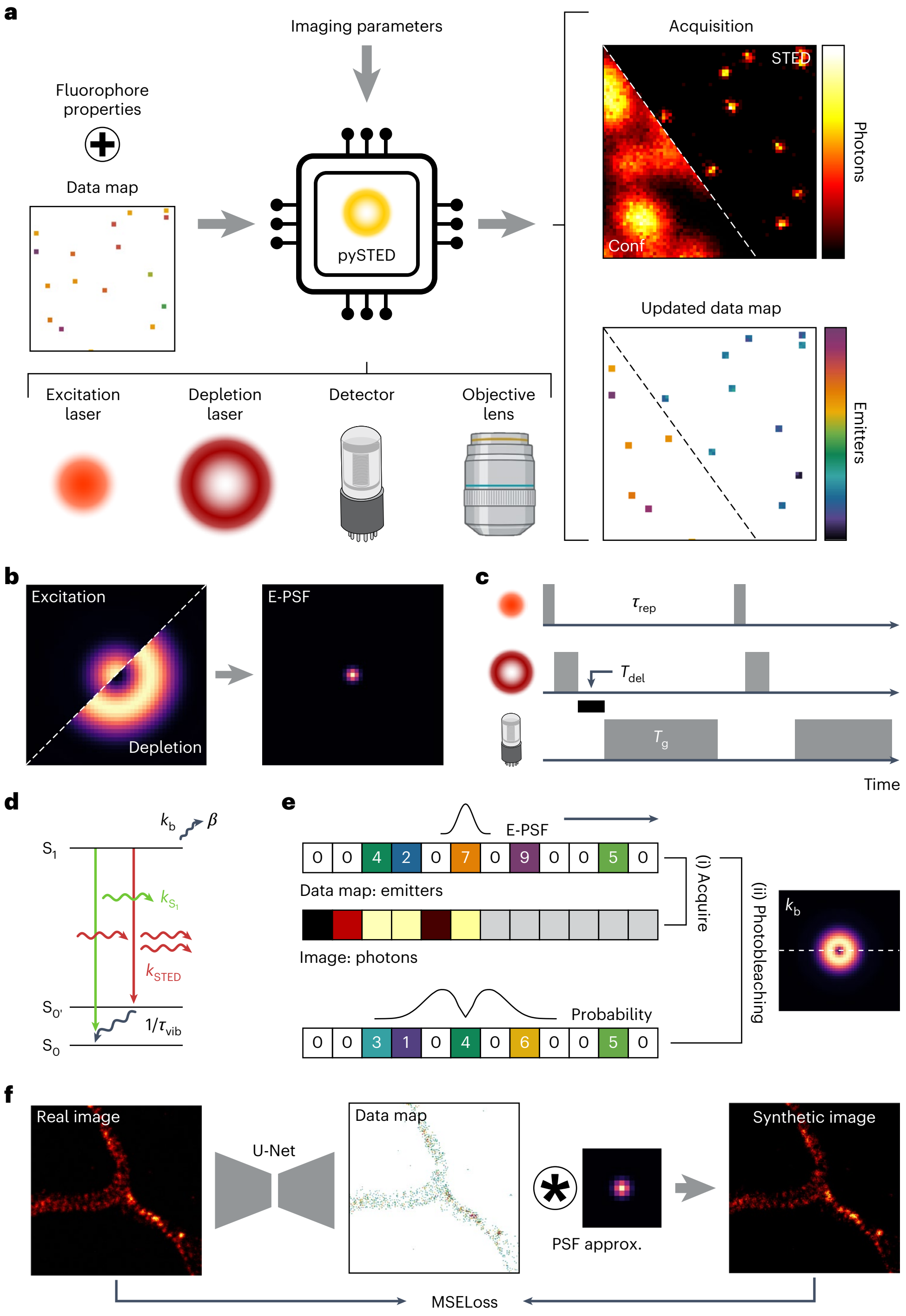


Fig. 1 | pySTED simulation platform. **a**, Schematic of the pySTED microscopy simulation platform. The user specifies the fluorophore properties (for example, brightness and photobleaching) and the positions of emitters in the data map. A simulation is built from several components (excitation and depletion lasers, detector and objective lens) that can be configured by the user. A low-resolution (Conf) or high-resolution (STED) image of a data map is simulated using the provided imaging parameters. The number of fluorophores at each location in the data map is updated according to their photophysical properties and associated photobleaching effects. **b**, Modulating the excitation with the depletion beam impacts the effective PSF (E-PSF) of the microscope. **c**, Time-gating module is implemented in pySTED, which affects the lasers and detection unit. The time-gating parameters of the simulation (gating delay, T_{del}; gating time, T_g) as well as the repetition rate of the lasers (τ_{rep}) are presented. A grey box is used to indicate when a component is active. **d**, Two-state Jablonski diagram (ground state, S_0; excited state, S_1) presents the transitions that are included in the fluorescence (spontaneous decay, k_{S_1}; stimulated emission decay, k_{STED}) and photobleaching dynamics (photobleaching rate, k_b; photobleached state, β) of pySTED. The vibrational relaxation rate ($1/\tau_{vib}$) affects the effective saturation factor in STED. **e**, Image acquisition is simulated as a two-step process at each location. Acquire (i): convolution of the E-PSF with the number of emitters in the data map (Data map: emitters) is calculated to obtain the signal intensity (Image: photons). Photobleaching (ii): number of emitters at each position in the data map is updated according to the photobleaching probability (line profile from k_b, compare the top and bottom lines). The same colour maps as in **a** are used. **f**, Realistic data maps are generated from real images. A U-Net model is trained to predict the underlying structure from a real STED image. Convolving the predicted data map with the approximated PSF results in a realistic synthetic image. During training, the mean squared error loss (MSELoss) is calculated between the real and synthetic image. Once trained, the convolution step can be replaced by pySTED. Objective lens in panel **a** created with BioRender.com.

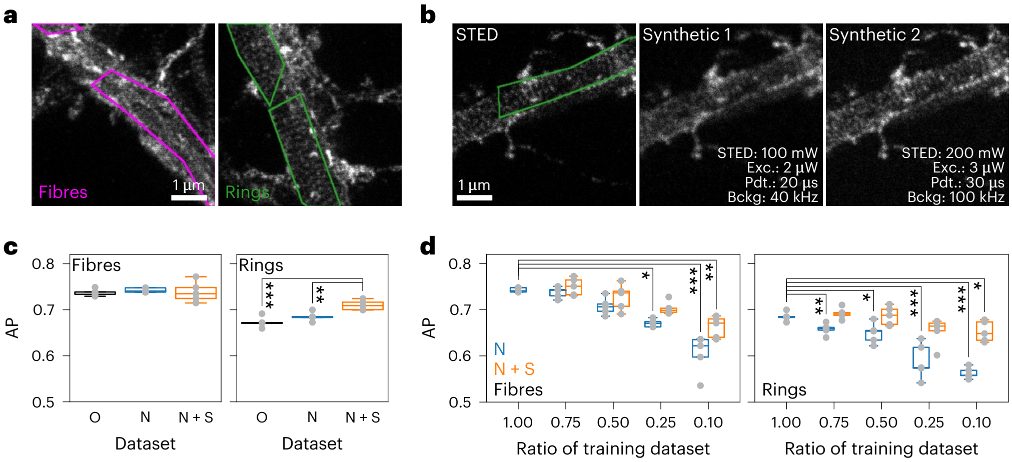


Fig. 2 | pySTED is used to artificially augment the training dataset of a DL model. a, Segmentation task used in ref. 46 is used, in which the annotations comprise polygonal bounding boxes around F-actin fibres (magenta) and rings (green). **b**, pySTED is used to augment the training dataset by generating synthetic versions of a STED image. **c**, AP of the model for the segmentation of F-actin fibres (magenta) and rings (green). The model was trained on the original dataset from ref. 46 (O), and on the same dataset with updated normalization (N) and additional synthetic images (N + S). No significant changes in AP are measured for F-actin fibres, but a significant increase is measured for N + S over O and N for F-actin rings (Supplementary Fig. 6 shows the *P* values). **d**, Images were progressively removed from the dataset (100%, 42 images; 75%, 31 images; 50%, 21 images; 25%, 10 images; and 10%, 4 images). Removing more than 50% of the dataset for fibres negatively impacts the models, whereas removing 25% of the dataset negatively impacts the segmentation of rings (N; Supplementary Fig. 6 shows the *P* values). Adding synthetic images from pySTED during training allows 75% of the original training dataset to be removed without affecting the performance for both structures (N + S; Supplementary Fig. 6 shows the *P* values). Only the significant changes from the complete dataset are highlighted. The complete statistical analysis is provided in Supplementary Fig. 6. All the box plots show the distribution of five model training scenarios. The box extends from the first to the third quartile of the data, with a line at the median. The whiskers extend from the box to the farthest data point lying within 1.5× the interquartile range (IQR) from the box.

We first compared how the imaging parameters on real microscopes and in pySTED simulations (pixel dwell time, excitation power and depletion power) influenced the image properties by measuring the resolution[40] and signal ratio[6] (Methods and Supplementary Fig. 3a). As expected, modulating the STED laser power influences the spatial resolution in real experiments and in pySTED simulations. Examples of the acquired and synthetic images are displayed in Supplementary Fig. 3b for visual comparison with different parameter combinations (Supplementary Fig. 3a). The impact of the imaging parameters in the resolution and signal ratio metrics in pySTED agree with the measurements that were performed on a real microscope. The small deviations can be explained by the variability that is typically observed in the determination of absolute values of fluorophore properties[41].

Next, we validated the photobleaching model that is implemented within pySTED. We calculated the photobleaching by comparing the fluorescence signal in a low-resolution image acquired before (CONF1) and after (CONF2) the high-resolution acquisition[6] (Methods). For the pixel dwell time, excitation power and other parameters (gating delay, gating time and line repetitions), we measured similar trends between real and synthetic image acquisitions (Supplementary Figs. 4a and 5). For a confocal acquisition, the photobleaching in pySTED is assumed to be 0 (Supplementary Fig. 4a), as it is generally negligible in a real confocal acquisition. Considering the flexibility of pySTED, different photobleaching dynamics specifically tailored for any particular experiment can be implemented and added in the simulation platform. Examples of sequential acquisition (ten images) are presented in Supplementary Fig. 4b, demonstrating the effect of imaging parameters on the photobleaching of the sample. pySTED also integrates background effects that can influence the quality of the acquired images as in real experiments[42,43] (Supplementary Fig. 4c,d).

`pySTED` as a development platform for AI-assisted microscopy

Dataset augmentation for training DL models. DL models are powerful tools to rapidly and efficiently analyse large databanks of images and perform various tasks such as cell segmentation[36,44]. When no pretrained models are readily available online to solve the task[45], fine tuning or training a DL model from scratch requires the tedious process of annotating a dataset. Here we aim to reduce the required number of distinct images for training by using pySTED as an additional data augmentation step. As a benchmark, we used the F-actin segmentation task from ref. 46, where the goal is to segment dendritic F-actin fibres or rings using a small dataset (42 images) of STED images (Fig. 2a and Methods). pySTED was used first as a form of data augmentation to increase the number of images in the training dataset without requiring new annotations. Using U-Net$_{\text{data map}}$, we generated F-actin data maps and a series of synthetic images in pySTED with various simulation parameters (Fig. 2b and Supplementary Table 8).

We compared the segmentation performance by using the average precision (AP; Methods) of a DL model trained on the original dataset (O, ref. 46) or with different image normalization and increased data augmentation (N). The segmentation performance was not impacted by increasing the amount of data augmentation (O versus N; Fig. 2c). Adding synthetic images from pySTED (N + S) into the training dataset to improve the diversity of the dataset significantly increases the performance of F-actin rings segmentation compared with O and N, and maintains the performance for the F-actin fibres segmentation (Fig. 2c). In biological experiments, where each image is costly to acquire, reducing the size of the training dataset results in a higher number of images for the post hoc analysis. Hence, we sought to measure the impact of reducing the number of real images in the training dataset by training on subsets of images that are augmented using pySTED (Supplementary Fig. 7). We measure a significant decrease in the AP for F-actin fibres when the model is trained on less than 50% of the images. Removing 25% of the dataset negatively impacts the segmentation performance of F-actin rings (Fig. 2d; Supplementary Fig. 6 shows the *P* values). However, adding synthetic images from pySTED during training allows the segmentation performance of the model to be maintained by training with only 25% of the original dataset (Fig. 2d; Supplementary Fig. 6 shows the *P* values).

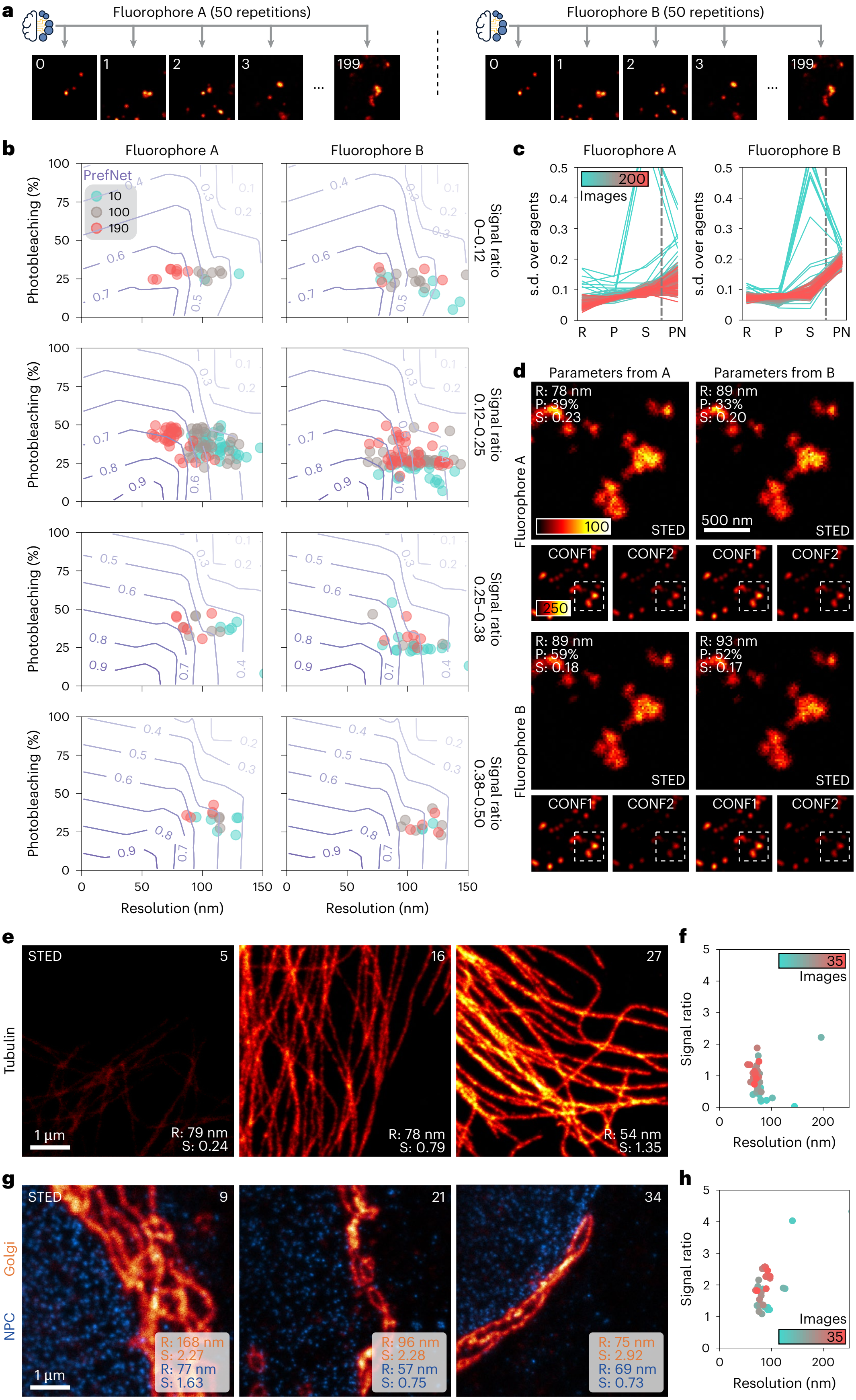

a
Fluorophore A (50 repetitions)
0
1
2
3
...
199
Fluorophore B (50 repetitions)
0
1
2
3
...
199
b
Fluorophore A
Fluorophore B
PrefNet
10
100
190
Photobleaching (%)
Signal ratio 0–0.12
Signal ratio 0.12–0.25
Signal ratio 0.25–0.38
Signal ratio 0.38–0.50
Resolution (nm)
0
50
100
150
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
c
Fluorophore A
Fluorophore B
200
Images
s.d. over agents
0.5
0.4
0.3
0.2
0.1
0
R
P
S
PN
d
Parameters from A
Parameters from B
Fluorophore A
R: 78 nm
P: 39%
S: 0.23
R: 89 nm
P: 33%
S: 0.20
100
500 nm
STED
CONF1
CONF2
250
Fluorophore B
R: 89 nm
P: 59%
S: 0.18
R: 93 nm
P: 52%
S: 0.17
STED
CONF1
CONF2
e
Tubulin
STED
5
16
27
1 µm
R: 79 nm
S: 0.24
R: 78 nm
S: 0.79
R: 54 nm
S: 1.35
f
35
Images
Signal ratio
Resolution (nm)
0
100
200
0
1
2
3
4
5
g
Golgi
NPC
STED
9
21
34
1 µm
R: 168 nm
S: 2.27
R: 77 nm
S: 1.63
R: 96 nm
S: 2.28
R: 57 nm
S: 0.75
R: 75 nm
S: 2.92
R: 69 nm
S: 0.73
h
Images
35
Signal ratio
Resolution (nm)
0
100
200
0
1
2
3
4
5

Fig. 3 | Validation of AI-assisted algorithms with pySTED for STED microscopy parameter optimization. a, pySTED is used to confirm the robustness of a model to the random initialization by repeatedly optimizing (50 repetitions) the imaging parameters on the same sequence of data maps (200 images). Two fluorophores are considered for demonstration purposes (Supplementary Table 10). **b**, Resulting imaging optimization objectives from LinTSDiag at three different time steps (10, cyan; 100, grey; 190, red) for 50 independent models, which are presented for increasing signal ratio (top to bottom). With time, LinTSDiag acquires images that have a higher preference score for both fluorophores (purple contour lines) and converges into a similar imaging optimization objective space (red points). **c**, Standard deviation (s.d.) of the imaging optimization objectives and of the preference scores decreases during optimization (cyan to red), supporting the convergence of LinTSDiag in a specific region of the imaging optimization objective space for both fluorophores. The dashed line separates the imaging optimization objectives (R, resolution; P, photobleaching; S, signal ratio) from the preference network (PN). **d**, Typical pySTED simulations on two different fluorophores (top/bottom) using the optimized parameters on fluorophore A (left) or B (right). Parameters that were optimized for fluorophore A (top left) result in higher photobleaching and maintain a similar resolution and signal ratio on fluorophore B (bottom left) compared with parameters that were optimized for fluorophore B (bottom right). Supplementary Table 10 lists the imaging parameters. **e**, Example acquisition of LinTSDiag of tubulin in kidney epithelial cells (Vero cells) stained with STAR RED in the beginning (left) and end (right) of optimization. **f**, Over time, LinTSDiag manages to increase both resolution and signal ratio of the acquired images (35 images, cyan to red). **g**, LinTSDiag allows multicolour imaging due to its high-dimensional parameter space capability. LinTSDiag optimizes the averaged resolution and signal ratio from both channels in dual-colour images acquired of the golgi (STAR ORANGE) and NPC (STAR RED) in Vero cells. **h**, LinTSDiag can maximize the signal ratio in the images and maintain the resolution of the images (35 images, cyan to red).

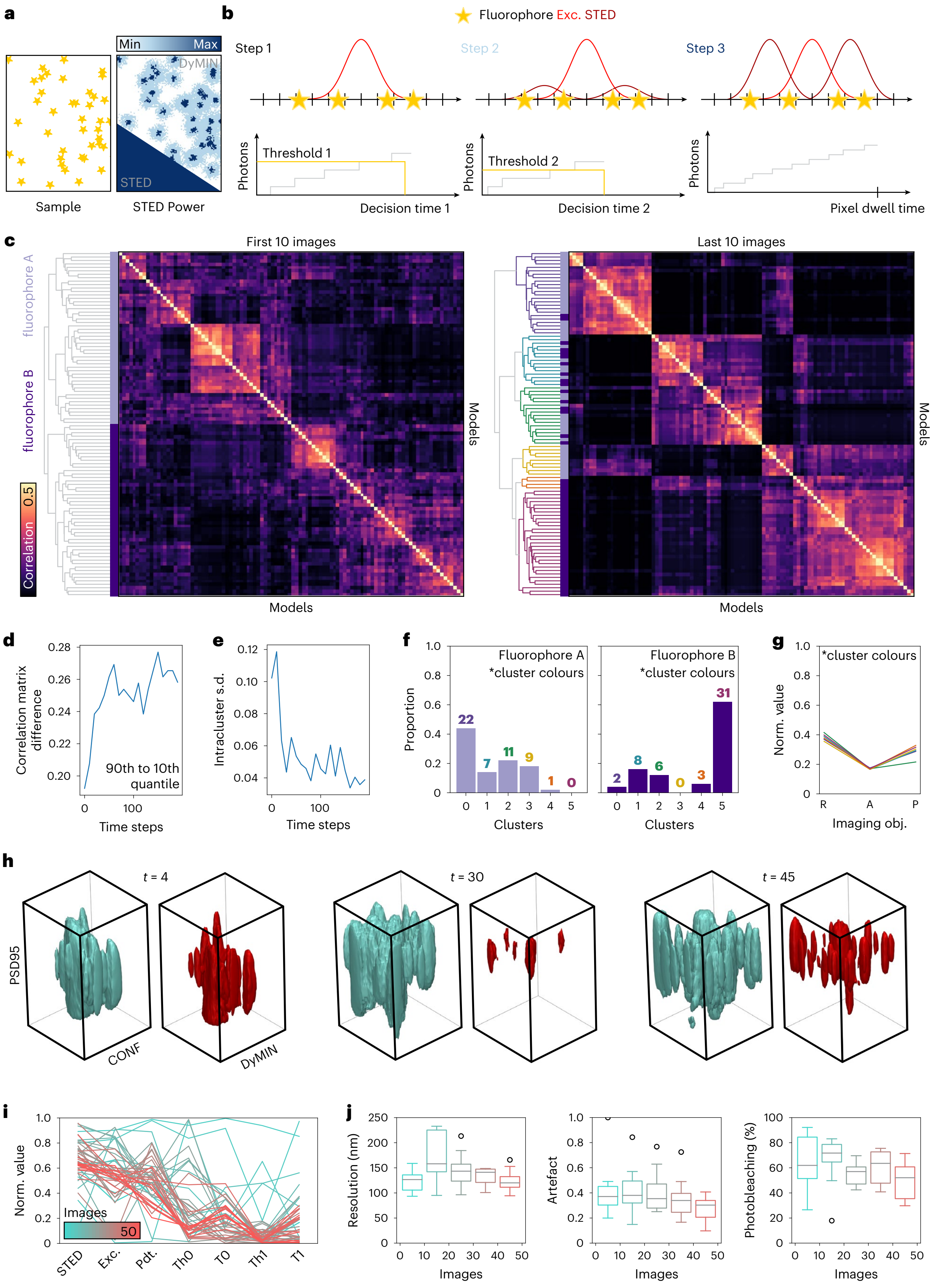
Validation of AI methods. Benchmarking AI models for automating microscopy tasks on biological samples is challenging due to biological variability and the difficulty of comparing imaging strategies on the same region of interest[6,22,47]. Assessing and comparing AI models requires multiple attempts in similar but different experimental conditions to limit the impact of biological variability. This inevitably increases the number of biological samples and the time required to develop robust AI-assisted adaptive microscopy strategies that can be deployed on a variety of samples and imaging conditions. pySTED allows the simulation of multiple versions of the same images as if the structure had been imaged with different experimental settings. Here we showcase the capability of pySTED in thoroughly validating ML approaches for the optimization of STED imaging parameters in a simulated controlled environment, enabling more robust performance assessments and comparisons.

We first demonstrate how pySTED can be used to characterize the performance of a multi-armed bandit optimization framework that uses Thompson sampling (TS) for exploration, that is, Kernel-TS. The application of Kernel-TS for the optimization of STED imaging parameters was demonstrated previously, but comparison between different experiments was challenging due to local variations in the size, brightness and photostability of the fluorescently tagged neuronal structures[6]. Using synthetic images generated with pySTED allows the performance of Kernel-TS to be evaluated on the same image sequence (50 repetitions; Methods) and with controlled photophysical properties of fluorophores (Extended Data Fig. 2 and Supplementary Tables 9 and 10). For experimental settings such as multichannel imaging or adaptive scanning, Kernel-TS is limited by the number of parameters that can be simultaneously optimized (~4) in an online setting[6]. We, thus, turned to a neural network implementation of TS, which was recently developed to solve the multi-armed bandit framework, LinTSDiag[48].

Using pySTED, we could characterize the performance of LinTSDiag on a microscopy optimization task on synthetic images without requiring real biological samples. As described above, LinTSDiag was trained on the same sequence (50 repetitions; Methods) using two different fluorophores (Fig. 3a and Supplementary Table 9). In a simple three-parameter optimization setting, LinTSDiag allows a robust optimization of the signal ratio, photobleaching and spatial resolution for fluorophores with distinct photophysical properties (Fig. 3b). We evaluate the performance of LinTSDiag using the preference score, which is obtained from a network that was trained to predict the preferences of an expert in the imaging optimization objective space (PrefNet; Methods)[6]. The convergence of the agent in the imaging optimization objective space is supported by the smaller standard deviation measured in the last iterations of the imaging session (Fig. 3c, red lines). pySTED enables a comparison of the optimized parameters for different fluorophores on the same data map. This experiment confirms that optimal parameters vary depending on the photophysical properties (Fig. 3d).

LinTSDiag was then deployed on a real microscopy system to simultaneously optimize four parameters (excitation power, STED power, pixel dwell time and line steps) for the imaging of tubulin stained with STAR RED in kidney epithelial cells (Vero cell line). The model was able to optimize the imaging optimization objectives, improve the resolution and signal ratio, and maintain a low level of photobleaching over the course of optimization (Fig. 3e,f and Supplementary Table 9). Then, we sought to increase the number of parameters by tackling a dual-colour imaging scheme (six parameters: excitation power, STED power and line steps for both channels) for the STED imaging of Golgi stained with STAR ORANGE and nuclear pore complex (NPC) stained with STAR RED in Vero cells (Fig. 3g,h and Supplementary Table 9). The optimization framework allows four imaging optimization objectives to be simultaneously optimized (for example, resolution and signal

Fig. 4 | Validation of contextual bandit algorithms with pySTED in a high-dimensional parameter space. a, DyMIN microscopy minimizes the light dose by using thresholds to turn off the high-intensity STED laser when no structure is within the doughnut beam (white regions). **b**, DyMIN uses a three-step process at each pixel: two thresholds (Thresholds 1 and 2) and two decision times (Decision Times 1 and 2) to turn off the lasers when unnecessary, followed by a normal STED acquisition (Step 3). **c**, pySTED characterizes LinTSDiag models optimizing seven parameters (STED power, excitation power, pixel dwell time, thresholds 1 and 2, and decision times 1 and 2) using prior task information (confocal image). Convergence of models is evaluated by measuring correlation in action selection (50 models) over time (Supplementary Fig. 8). Clustering of the correlation matrix reveals clusters of policies that are better defined later in the optimization process (right dendrogram, colour coded). Shades of purple on the left represent two fluorophores (light, A; dark, B). **d**, Difference between the 90th and 10th quantiles of the correlation matrix increases with time, implying better-defined clusters of policies. **e**, Intracluster s.d. of parameter selection decreases, showing policy convergence in all the clusters. **f**, Proportion of models per cluster for fluorophore A or B (light and dark, respectively) shows different modes of attraction in the parameter space for fluorophores with distinct photophysical properties (colour code from **c**). **g**, Although models converged in different regions of parameter space, the measured imaging optimization objectives (A, artefact) are similar for each cluster (colour code from **c**). **h**, Example of real acquisition with LinTSDiag optimization for DyMIN3D of the synaptic protein PSD95 in cultured hippocampal neurons shows confocal (left) and DyMIN (right) (size, 2.88 μm × 2.88 μm × 2 μm). **i**, Parameter selection convergence in the seven-parameter space is observed (cyan to red; STED, STED power; Exc., excitation power; Pdt., pixel dwell time; Th1–2, DyMIN thresholds; and T1–2, DyMIN decision times). **j**, LinTSDiag optimization reduces the variability in imaging optimization objectives during the optimization (50 images). The box plot shows distribution in bins of 10 images for 50 repetitions, with boxes extending from the first to the third quartile and whiskers extending to the farthest data point within 1.5× IQR.


a
Min
Max
DyMIN
STED
Sample
STED Power
b
Fluorophore Exc. STED
Step 1
Step 2
Step 3
Photons
Threshold 1
Decision time 1
Photons
Threshold 2
Decision time 2
Photons
Pixel dwell time
c
First 10 images
Last 10 images
fluorophore A
fluorophore B
Correlation
0.5
Models
Models
Models
Models
d
Correlation matrix difference
0.28
0.26
0.24
0.22
0.20
90th to 10th quantile
0
100
Time steps
e
Intracluster s.d.
0.12
0.10
0.08
0.06
0.04
0
100
Time steps
f
Proportion
1.0
0.8
0.6
0.4
0.2
0
Fluorophore A
*cluster colours
22
7
11
9
1
0
0 1 2 3 4 5
Clusters
Fluorophore B
*cluster colours
2
8
6
0
3
31
0 1 2 3 4 5
Clusters
g
Norm. value
1.0
0.8
0.6
0.4
0.2
0
*cluster colours
R
A
P
Imaging obj.
h
t = 4
t = 30
t = 45
PSD95
CONF
DyMIN
i
Norm. value
1.0
0.8
0.6
0.4
0.2
0
Images
50
STED
Exc.
Pdt.
Th0
T0
Th1
T1
j
Resolution (nm)
250
200
150
100
50
0
0 10 20 30 40 50
Images
Artefact
1.0
0.8
0.6
0.4
0.2
0
0 10 20 30 40 50
Images
Photobleaching (%)
100
80
60
40
20
0
0 10 20 30 40 50
Images

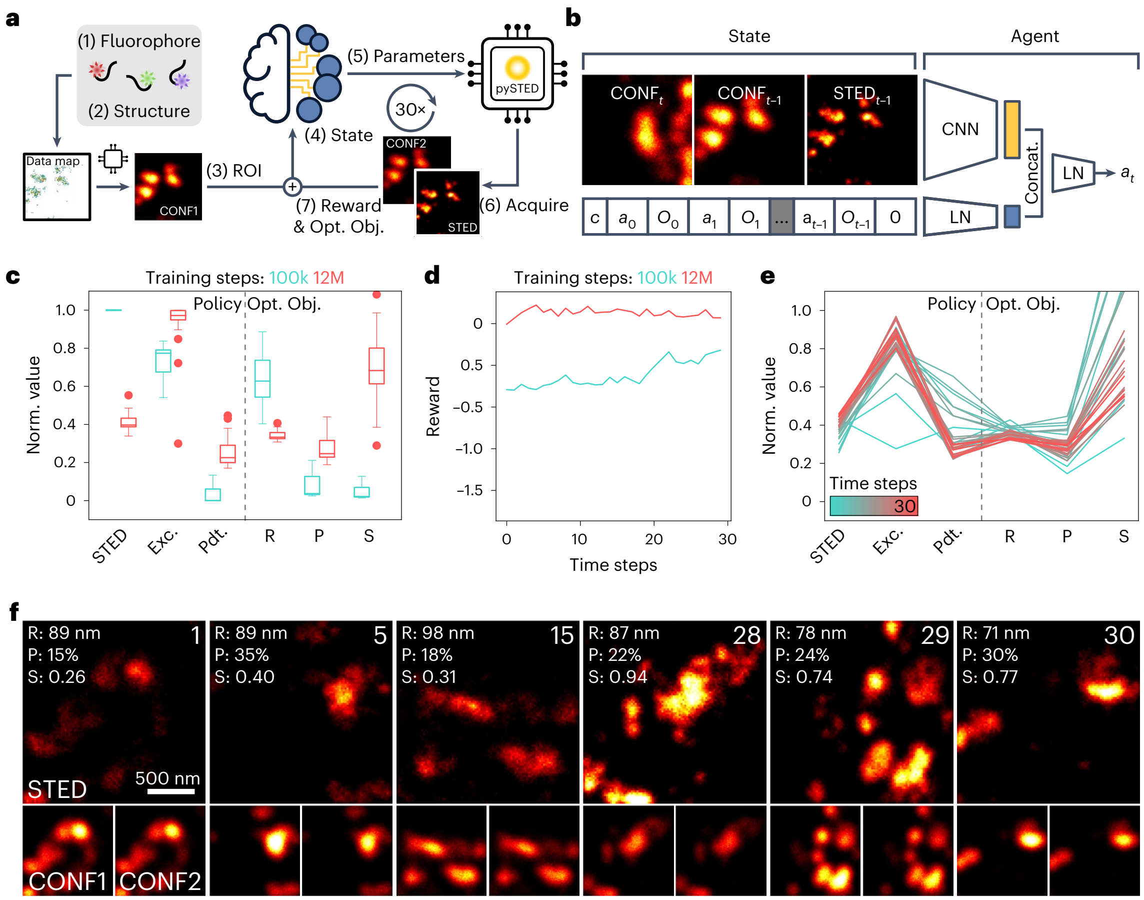


Fig. 5 | RL agent training for STED imaging parameter optimization with pySTED . **a**, RL training loop: each episode starts by sampling the photophysical properties of a fluorophore (1) and selecting a structural protein from the databank (2). At each time step, a region of interest (ROI) is selected: a data map is created and a confocal image is generated with pySTED (3). This image is used in the state of the agent (4) to select the subsequent imaging parameters (5). A STED image and a second confocal image are generated in pySTED (6). The imaging optimization objectives and the reward are calculated (Reward & Opt. Obj.) (7). **b**, State of the agent includes a visual input (current confocal (CONF$_t$) and the previous confocal/STED images (CONF$_{t-1}$ and STED$_{t-1}$)), and incorporates the laser excitation power of the confocal image (*c*), history of actions (a_t) and imaging optimization objectives (O_t). The history is zero padded to a fixed length (0). Visual information is encoded using a CNN, and the history, using a fully connected LN. Both encodings are concatenated and fed to an LN model to predict the next action. **c**, Evolution of the policy (left) and imaging optimization objectives (right) for a fluorophore with high-signal, low-photobleaching properties at the start (cyan; 100k time steps) and end (red; 12M time steps) of the training. A box plot shows the distribution of the average value from the last 10 images of an episode (30 repetitions), with boxes extending from the first to the third quartile and whiskers extending to the farthest data point within 1.5× IQR. **d**, Evolution of the reward during an episode at the beginning (cyan; 100k time steps) and end (red; 12M time steps) of training. **e**, Evolution of the policy (left) and imaging optimization objectives (right) after training (12M time steps) during an episode. **f**, Typical images acquired during an episode (top right, image index; top left, imaging optimization objectives). The STED and second confocal (CONF2) image are normalized to their first confocal (CONF1) images. For the data in **c**–**f**, the same fluorophore is used.

ratio for both colours). As the visual selection of the trade-off in a four-dimensional space is challenging for the user in an online setting, we decided to optimize the combined resolution and signal ratio of both fluorophores (average of the imaging optimization objectives), allowing the users to indicate their preference in a two-dimensional optimization objective space. Online six-parameter optimization of LinTSDiag increases the signal ratio and maintains good image resolution for both imaging channels (Fig. 3h), enabling the resolution of both structures with sub-100-nm resolution.

Next, we developed a model that leverages prior information (context) to solve a task with a high-dimensional action space. This is the case for dynamic intensity minimum (DyMIN) microscopy that requires parameter selection to be adapted, particularly multiple illumination thresholds, to the current region of interest[8] (Fig. 4a,b). We previously showed that contextual bandit algorithms can use the confocal image as a context to improve the DyMIN threshold optimization in a two-parameter setting[49]. In this work, we aim to increase the number of parameters (seven parameters) that can be simultaneously optimized and validate the robustness of LinTSDiag[48] (Fig. 4b). We repeatedly trained LinTSDiag on the same data map sequence using the confocal image as prior information (50 repetitions). The parameter selection was compared by measuring whether the action selection correlated over time between the models (Fig. 4c, Supplementary Fig. 8, Supplementary Table 9 and Methods). For instance, the correlation matrix from the last ten images shows clusters of similar parameters that are better defined than for the first ten images (Fig. 4c). This is confirmed by the difference in the 90th and 10th quantiles in the correlation matrix, which rapidly increases with time (Fig. 4d). As expected with clustered policies, the average standard deviation of the action selection for each cluster reduces over time, implying similar parameter selection by the models (Fig. 4e). We also assessed whether the models would adapt their policies to different fluorophores (Fig. 4c,f, light/dark purple). As shown in Fig. 4f, there are specific policies for each fluorophore (for example, fluorophore A, 0 and 3; fluorophore B, 5), demonstrating the capability of the models in adapting their parameter selections to the experimental condition. Although the policy of the models is different, the measured imaging optimization objectives are similar for all the clusters (Fig. 4g), which suggests that different policies can solve this task, unveiling the delicate intricacies of DyMIN microscopy. More importantly, this shows that the model can learn one of the many possible solutions to optimize the imaging task.

The LinTSDiag optimization strategy was deployed in a real-life experiment for the seven-parameter optimization of DyMIN3D imaging of the post-synaptic protein PSD95 in dissociated primary hippocampal neurons stained with STAR 635P. Early during the optimization, the selected parameters produced images with poor resolution or missing structures (artefacts) (Fig. 4h and Supplementary Table 9). The final images were of higher quality (Fig. 4h, right) with fewer artefacts and high resolution. The parameter selection of the model converged in a region of parameter space that could improve all the imaging optimization objectives over the course of optimization (Fig. 4i,j). Parameters optimized with LinTSDiag allowed a significant improvement in the DyMIN3D imaging of PSD95 compared with conventional three-dimensional STED imaging (Supplementary Fig. 9). pySTED allowed us to validate the robustness of the model in a simulated environment before its deployment in a real experimental setting. This should benefit the ML community by allowing the validation of new online ML optimization algorithms on realistic tasks.

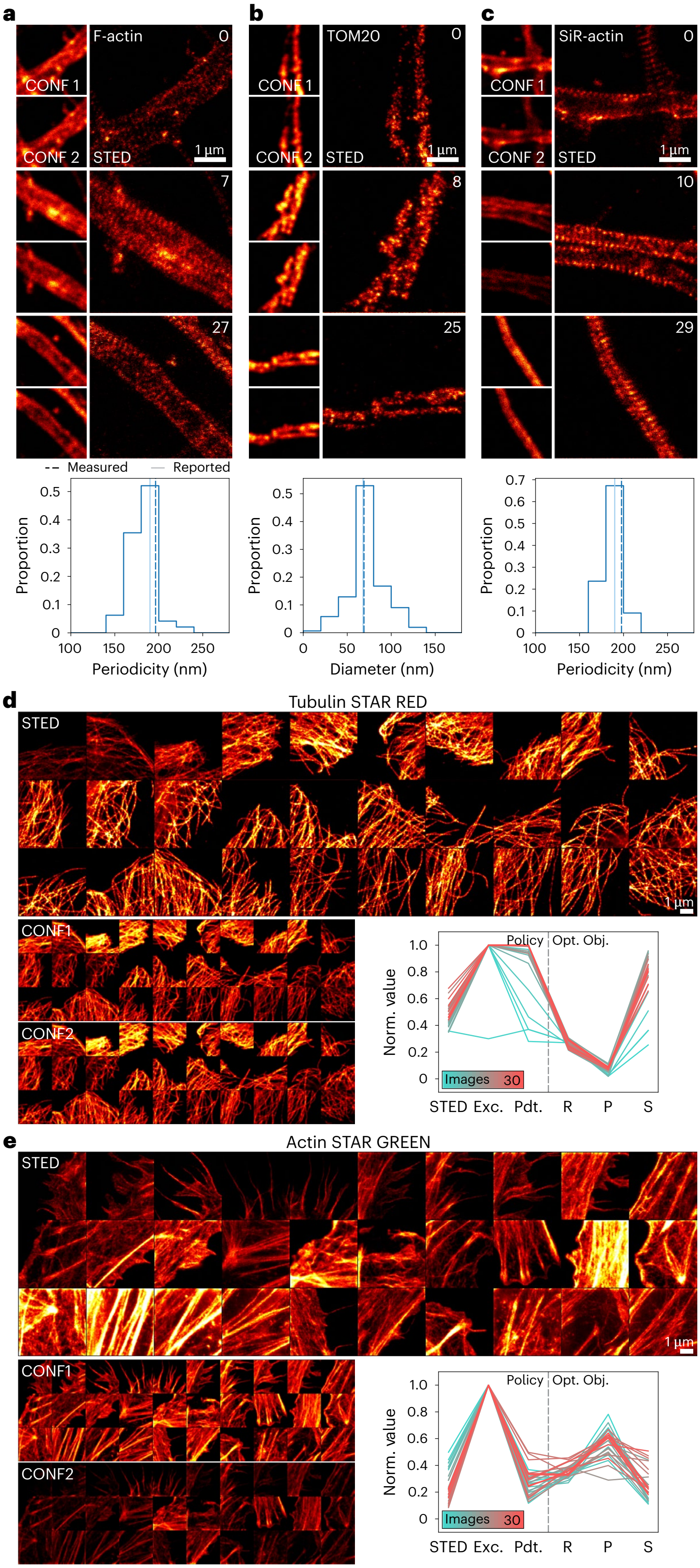
Learning through interactions with the system

Online optimization strategies such as Kernel-TS and LinTSDiag were trained from scratch on a new sample, implying a learning phase in which only a fraction of the images meet the minimal image quality requirements. For costly biological samples, there is a need to deploy algorithms that can make decisions based on the environment with a reduced initial exploration phase. Control tasks and sequential planning are particularly well suited for an RL framework where an agent (for example, replacing the microscopist) learns to make decisions by interacting with the environment (for example, select imaging parameters on a microscope) with the aim of maximizing a reward signal (for example, light exposure, signal ratio and resolution) over the course of an episode (for example, imaging session)[50]. Deep RL agents are (unfortunately) famously data intensive, sometimes requiring millions of examples to learn a single task[17,30]. This makes them less attractive to be trained on real-world tasks in which each sample can be laborious to obtain (for example, biological samples) or when unsuitable actions can lead to permanent damage (for example, overexposition of the photon detector). Simulation platforms are, thus, essential in RL to provide environments in which an agent can be trained at low cost for subsequent deployment in a real-life scenario[51], which is referred to as simulation to reality (Sim2Real) in robotics. Although Sim2Real is widely studied in robotics and autonomous driving, its success for new fields of application is generally dependent on the gap between simulation and reality[52].

Here pySTED is used as simulation software to train RL agents. We implemented pySTED in an OpenAI Gym environment (gym-STED) to facilitate the deployment and development of RL strategies for STED microscopy[19,53]. To highlight the potential of gym-STED to train an RL agent, we crafted the task of resolving nanostructures in simulated data maps of various neuronal structures (Fig. 5a). In gym-STED, an episode unfolds as follows. At each time step, the agent observes the state of the sample: a visual input (image) and the current history (Methods and Fig. 5b). The agent then performs an action (adjusting pixel dwell time, excitation power and depletion power), receives a reward based on the imaging optimization objectives and transitions into the next state. A single-value reward is calculated using a preference network that was trained to rank the imaging optimization objectives (resolution, photobleaching and signal ratio) according to expert preferences[6] (Methods). A negative reward is obtained when the selected parameters lead to a high photon count that would be detrimental to the detector in real experimental settings (for example, nonlinear detection of photon counts). This sequence is repeated until the end of the episode, that is, 30 time steps. In each episode, the goal of the agent is to balance between detecting the current sample configuration and acquiring high-quality images to maximize its reward (Fig. 5a). We trained a proximal policy optimization (PPO)[54] agent and evaluated its performance on diverse fluorophores (Methods). Domain randomization is heavily used within the simulation platform to cover a wide variety of fluorophores and structures and thus increase the generalization properties of the agent[55]. In Fig. 5c–f (Supplementary Table 11), we report the performance of the agent on a fluorophore with simulated photophysical properties that would result in high brightness (high signal ratio) and high photostability (low photobleaching) in real experiments. The results of the agent on other simulated fluorophore properties are also reported (Supplementary Table 12 and Supplementary Fig. 10). Over the course of training, the agent adapts its policy to optimize the imaging optimization objectives (100k and 12M training steps; Fig. 5c). As expected from RL training, the reward of an agent during an episode is greater at the end of training compared with the beginning (Fig. 5d, red versus cyan). When evaluated on a new sequence, the agent trained over 12M steps rapidly adapts its parameter selection during the episode to acquire images with high resolution and signal ratio and minimize photobleaching (Fig. 5e,f). The agent shows a similar behaviour for various simulated fluorophores (Supplementary Fig. 10). We compared the number of good images acquired by the RL agent with that of bandit optimization for the first 30 images of the optimization. In similar experimental conditions, with the same fluorophore and parameter search space, the average number of good images was (18 ± 3) and (5 ± 3) for the RL agent and bandit, respectively (50 repetitions). This almost fourfold increase in the number of high-quality images highlights the improved efficiency of the RL agent at suggesting optimal imaging parameters.

Given the capability of the agent in acquiring images for a wide variety of synthetic imaging sequences, we evaluated if the agent

Fig. 6 | Bridging the reality gap in RL by pretraining with pySTED. For real microscopy experiments, the agent was trained for over 12M steps in the simulation. It was then deployed on a real STED microscope to image diverse proteins in dissociated neuronal cultures and cultivated Vero cells. **a**, Top: simulated images of F-actin in fixed neurons were used during training. Deploying the RL agent to acquire an image of this in-distribution structure in a real experiment allows to resolve the periodic lattice of F-actin tagged with phalloidin-STAR635. Bottom: structural parameters extracted from the images (Methods; the dashed vertical line represents the median of the distribution) were compared with the literature value (solid vertical line), showing that the agent adjusted the imaging parameters to resolve the 190 nm periodicity of F-actin[56,57]. **b**, Top: the trained agent is tested on the protein TOM20, unseen during training (out of distribution). The nano-organization of TOM20 is revealed in all the acquired images. Bottom: the measured average cluster diameter of TOM20 concords with the averaged reported values from ref. 60. **c**, Top: live-cell imaging of SiR-actin shows the model's adaptability to different experimental conditions (out of distribution). Bottom: the periodicity of the F-actin lattice is measured and compared with the literature. In **a**–**c**, STED images are normalized to their respective confocal image (CONF1). The second confocal image (CONF2) uses the same colour scale as CONF1 to reveal the photobleaching effects. **d**,**e**, Images acquired by the RL agent on a different microscope of tubulin (**d**; STAR RED) and actin (**e**; STAR GREEN) in fixed Vero cells. Image sequences from top left to bottom right show confocal images before (CONF1) and after (CONF2) photobleaching, with CONF2 normalized to the CONF1 image. The STED images are normalized to the 99th percentile of the intensity of the CONF1 image. Images are 5.12 µm × 5.12 µm. The evolution of the parameter selection (left) and imaging optimization objectives (right) showed that optimal parameters and optimized objectives differed for STAR RED (**d**) and STAR GREEN (**e**).


a
F-actin
0
CONF 1
CONF 2
STED
1 μm
7
27
b
TOM20
0
CONF 1
CONF 2
STED
1 μm
8
25
c
SiR-actin
0
CONF 1
CONF 2
STED
1 μm
10
29
-- Measured
— Reported
Proportion
0
0.1
0.2
0.3
0.4
0.5
100
150
200
250
Periodicity (nm)
Proportion
0
0.1
0.2
0.3
0.4
0.5
0
50
100
150
Diameter (nm)
Proportion
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
100
150
200
250
Periodicity (nm)
d
Tubulin STAR RED
STED
1 μm
CONF1
CONF2
Policy
Opt. Obj.
Norm. value
1.0
0.8
0.6
0.4
0.2
0
Images 30
STED
Exc.
Pdt.
R
P
S
e
Actin STAR GREEN
STED
1 μm
CONF1
CONF2
Policy
Opt. Obj.
Norm. value
1.0
0.8
0.6
0.4
0.2
0
Images 30
STED
Exc.
Pdt.
R
P
S

could be deployed in a real experimental setting. The experimental conditions chosen for the simulations were based on the parameter range available on the real microscope. Dissociated primary hippocampal neurons were stained for various neuronal proteins (Fig. 6, Extended Data Fig. 3 and Supplementary Table 11) and imaged on a STED microscope with the RL agent assistance for parameter choice. First, we evaluated the performance of our approach for Sim2Real on in-distribution images from F-actin and CaMKII-β in fixed neurons. Although the simulated images of both structures were available within the training environment, we wanted to evaluate if the agent could adapt to real-life imaging settings (Supplementary Fig. 11). As shown in Fig. 6a and Extended Data Fig. 3, the agent resolves the nano-organization of both proteins (Supplementary Fig. 11). We sought to confirm whether the quality of the images was sufficient to extract biologically relevant features (Methods). For both proteins, the measured quantitative features matched with values previously reported in the literature, enabling the resolution of the 190 nm periodicity of the F-actin lattice in axons, and the size distribution of CaMKII-β nanoclusters[56–58] (Fig. 6a and Extended Data Fig. 3). Next, we wanted to validate that the agent would adapt its parameter selection to structures, fluorophores properties or imaging conditions that were not included in the training set. We first observed that the agent could adapt to a very bright fluorescent signal and adjust the parameters to limit the photon counts on the detector (Extended Data Fig. 3). The morphology of the imaged PSD95 nano-cluster was in agreement with the values reported in another work[59] (Extended Data Fig. 3). We deployed the RL-based optimization scheme for the imaging of the mitochondrial protein TOM20 to evaluate the ability of the agent to adapt to out-of-distribution structures (Fig. 6b). The nano-organization and morphology described previously[60] for TOM20 in punctate structures is revealed using the provided imaging parameters in all the acquired images (Fig. 6b and Supplementary Fig. 11). Next, we evaluated the generalizability of the approach to a new imaging context, which is live-cell imaging. We used the optimization strategy for the imaging of the F-actin periodic lattice in living neurons (Fig. 6c). The quality of the acquired images is confirmed by the quantitative measurement of the periodicity, which matches the previously reported values of 190 nm from the literature[56,57]. Finally, we verified the generalizability of our approach by deploying our RL-assisted strategy on a new microscope and samples (Fig. 6d,e, Extended Data Fig. 4 and Supplementary Table 11). We evaluated the performance of the RL agent for imaging in fixed Vero cells of tubulin stained with STAR RED and actin stained with STAR GREEN. The agent successfully adapted to the new imaging conditions, rapidly acquiring high-quality images, even in challenging photobleaching conditions such as with STED microscopy of the green-emitting fluorophore STAR GREEN. Using the pySTED simulation environment, we could successfully train RL agents that can be deployed in a variety of real experimental settings to tackle STED imaging parameter optimization tasks. To the best of our knowledge, this is the first application of RL agents to an online image acquisition task in optical microscopy.

Discussion

We built pySTED, an in silico super-resolution STED environment, which can be used to develop and benchmark AI-assisted STED microscopy. Throughout the synthetic and real experiments, we have demonstrated that it can be used for the development and benchmarking of AI approaches in optical microscopy. The CoLaboratory notebook that was created as part of this work can be used by microscopist trainees to develop their skills and intuition for STED microscopy before using the microscope for the first time. The optimal set of parameters defined in pySTED for a specific fluorophore can guide the parameter choice on a real microscope, but should not replace optimization in real experimental settings to account for environmental effects and biological variability.

The simulation platform was built to be versatile and modular. This allows the users to create and test the efficiency of AI strategies and adaptive imaging schemes before deploying them on a real microscope. For instance, both DyMIN[8] and RESCue[61] microscopy techniques are readily available to the users. Additionally, the community can contribute open-source modules that would meet their experimental settings.

Smart microscopy requires the development of tools and modules to increase the capabilities of microscopes[10,62], which can be challenging when working on a real microscopy system. The development of simulation software is one way to mitigate the difficulty of building an AI-assisted microscopy setup. We mainly focused on the selection of imaging parameters, which is one branch of AI-assisted microscopy, but also showed that pySTED can be successfully applied to data augmentation in supervised learning settings. A recent trend in microscopy focuses on the implementation of data-driven microscopy systems. For example, systems are built to automatically select informative regions or improve the quality of the acquired images[63,64]. The development and validation of such data-driven systems could be achieved with pySTED. An interesting avenue to pursue for data-driven systems could rely on generative models to create diverse data maps on the fly instead of relying on existing databanks of STED microscopy images, which could be integrated into the modular structure of the pySTED simulation environment. Online ML optimization strategies tested in the pySTED environment showed similar performances when transferred to the real microscopy environment, opening new possibilities to characterize and benchmark innovative data-driven microscopy approaches in pySTED before their deployment on real biological samples.

We also tackle the training of an RL agent—the first for optical microscopy—which would be impossible without access to a large databank of simulated data. The RL agent enables a full automatization of the imaging parameter selection on a real system when deployed from gym-STED, an OpenAI gym environment built around pySTED[53]. Domain randomization was heavily used within the simulation platform[55], which resulted in an RL agent that could adapt its parameter selection to a wide variety of experimental conditions, even in living samples. Such strategies could be transformative to democratize STED microscopy to a larger diversity of experimental settings and allow non-expert users to acquire high-quality images on a new sample without previous optimization sessions.

Although RL agents can represent a powerful tool to automatize microscopy setups, they must be trained on a very large number of examples (for example, 12M steps in this work)[17,30], which would be infeasible on a real microscopy setup. The pySTED simulation environment allowed the RL agent to bridge the gap between simulation and reality without requiring any fine tuning. This makes pySTED an appealing platform for RL development as it is particularly well suited for complex control tasks requiring temporally distinct trade-offs to be made[20]. In this work, the model relied on a constant preference function to convert the multi-objective optimization into a single-reward function. This preference function is ultimately user dependent. This could be complemented in the future by incorporating RL from human feedback in the training of the RL model[65,66]. In future work, temporal dynamics could also be implemented in pySTED to open new possibilities to fully automatize the selection of informative regions and imaging parameters in an evolving environment.

Methods

pySTED simulation platform

Two main software implementations are incorporated within the pySTED simulation platform: (1) PSF calculation and (2) emitter–light interactions.

PSF calculation. PSF calculation in pySTED is inspired by previous theoretical work[26,29] (Fig. 1b). As in ref. 26, we calculate the excitation and depletion PSF by using the electric field (Fig. 1b). The E-PSF is calculated by combining the excitation, depletion and detection PSFs using

$$\text{E-PSF}\left(\vec{r}\right) = \text{PSF}_{\text{Exc}}\left(\vec{r}\right)\exp\left[-\text{PSF}_{\text{STED}}\left(\vec{r}\right)\zeta\right]\left[\text{PSF}_{\text{det}}\left(\vec{r}\right)\otimes \text{circ}(R)\right], \quad (1)$$

where R is the radius of the imaged aperture[67] and ζ is the saturation factor of the depletion defined as $\zeta = \frac{I_{\text{STED}}}{I_{\text{s}}}$, with I_{s} being the saturation intensity[29]. The left side of equation (1) represents the probability that an emitter at position $\vec{r}$ contributes to the signal[68] and is calculated in pySTED using ηp_{exc} with

$$p_{\text{exc}} = q_{\text{fl}}(1-\exp\left(-\sigma_{\text{abs}}\Phi_{\text{exc}}\tau_{\text{STED}}\right)), \quad (2)$$

where q_{fl} is the quantum yield of the fluorophore, σ_{abs} is the absorption cross-section, Φ_{exc} is the photon flux from the excitation laser and τ_{STED} is the period of the STED laser. The parameter η allows the excitation probability to be modulated with the depletion laser or allows time gating to be considered during the acquisition[29,69]. Time gating consists of activating the detector within a small window of time (T_{g}; typically 8 ns) after the excitation pulse (T_{del}; typically 750 ps) to prominently detect photons coming from spontaneous emission. The simulations performed with pySTED follow the scheme of pulsed STED microscopy in which time gating mostly reduces the correlated background[69]. Following the derivation from ref. 29 and assuming that $T_{\text{g}} \geq \tau_{\text{STED}}$, the emission probability of a fluorophore is described as

$$F(\gamma) = \exp\left(-k_{\text{S1}}\gamma t_{\text{STED}}\right)\left(\exp\left(-k_{\text{S1}}T_{\text{del}}\right)-\exp\left(-k_{\text{S1}}(T_{\text{del}}+T_{\text{g}})\right)\right), \quad (3)$$

where k_{S1} is the spontaneous decay rate, γ is the effective saturation factor ($\gamma = \frac{\zeta k_{\text{vib}}}{(\zeta k_{\text{S1}+k_{\text{vib}}})}$; k_{vib} is the vibrational relaxation state of $S_{0'}$) and t_{STED} is the STED pulse width (Fig. 1c,d). In the confocal case ($I_{\text{STED}} = 0$), the emission probability simply reduces to

$$F(0) = \left(1-\exp\left(-k_{\text{S1}}T\right)\right), \quad (4)$$

where T is the period between each STED pulse. This allows the probability of spontaneous decay η to be calculated using $F(\gamma)/F(0)$. The calculated E-PSF is convolved on the data map to simulate the photons that are emitted and the one measured by the detector.

In real experiments, the number of detected photons is affected by several factors (for example, photon detection and collection efficiency of the detector, detection PSF, fluorophore brightness and so on), which were also integrated in the pySTED simulation environment (Supplementary Tables 1–5). We also included the possibility to add typical sources of noise that occur in a targeted microscopy experiment such as shot noise, dark noise and background noise, all of which are modelled by Poisson processes (Supplementary Table 3).

Emitter–light interactions. In a real microscopy experiment, the emitters can be degraded as they interact with excitation or depletion light. Photobleaching is the process by which an emitter becomes inactive following light exposure[3]. In STED microscopy, this process is mainly caused by the combination of excitation and depletion laser beams[3]. Reducing photobleaching is an optimization objective that the microscopist has to target during an imaging session and that must be minimized to preserve the sample health and sufficient imaging contrast. Hence, we implemented a realistic photobleaching model within the pySTED simulation software. The photobleaching model is based on the derivations from ref. 3, which were validated on real samples. Fig. 1d presents the energy states, decay rates and photobleaching state β that are used within the photobleaching model.

As in ref. 3, we define the photobleaching rate as

$$k_{\beta}(I_{\text{STED}}) = k_0\left(\frac{I_{\text{STED}}}{1\,\text{W}\,\text{m}^{-2}}\right)+k_1\left(\frac{I_{\text{STED}}}{1\,\text{Wm}^{-2}}\right)^{b}, \quad (5)$$

where k_0, k_1 and b are dependent on the fluorophore and have to be determined experimentally. In the default parameters of pySTED, we assume that the linear photobleaching term is null ($k_0 = 0$) and that photobleaching occurs only from S1 during the STED pulse. Other photobleaching parameters could be easily integrated considering the modular structure of pySTED. We define the effective photobleaching rate k_{b} as the number of emitters transitioning from the S1 state to the photobleached state (P_{β}) over the course of a laser period T:

$$k_b = \frac{P_{\beta}}{T} \quad (6)$$

with

$$P_{\beta} = P_{\text{S1}}(t=0)k_{\beta}(I_{\text{STED}})\frac{1-\exp\left(-k_{\text{S1}}t_{\text{STED}}(1+\gamma)\right)}{k_{\text{S1}}(1+\gamma)}. \quad (7)$$

In pySTED, the number of emitters N in a pixel is updated by calculating their survival probability $p = \exp(-k_{\text{b}}t)$ from a binomial distribution for a given dwell time t (Fig. 1e). Although most parameters can be obtained from the literature for a specific fluorophore, some parameters such as k_1 and b need to be determined experimentally[3]. Given some experimental data (or a priori about the expected photobleaching of a sample), we can estimate the photobleaching properties (k_1 and b) of a fluorophore with

$$\text{Photobleaching} = 1-\exp\left(-k_{\text{b}}t\right), \quad (8)$$

by using nonlinear least-squares methods. We can also apply a similar process to estimate the absorption cross-section (σ_{abs}) of a fluorophore to optimize the confocal signal intensity to an expected value[3].

Realistic data maps

A realistic data map, which can be used in pySTED, is generated by predicting the position of emitters in a real super-resolved image. A U-Net model (U-Net$_{\text{data map}}$, implemented in PyTorch[70]) is trained to predict the underlying structure of a super-resolved image (Supplementary Table 13). A single U-Net$_{\text{data map}}$ was trained in this Article with images of different subcellular structures (F-actin, tubulin, PSD95, αCaMKII and LifeAct) and was used to generate all the data maps to train and validate the ML, DL and RL models presented in this study. U-Net$_{\text{data map}}$ has a depth of 4 with 64 filters in the first double convolution layer. Padding was used for each convolution layer to maintain the same image size. As in the seminal implementation of U-Net[35], maxpool with a kernel and stride of 2 was used. The number of filters in the double convolution layers doubled at each depth. In the encoder part of the model, each convolution is followed by batch normalization and a rectified linear unit (ReLU). Upsampling is performed using transposed convolution. The decoder part of the model uses double convolution layers as in the encoder part of the model. At each depth of the model, features from the encoder are propagated using skipping links and concatenated with the features obtained following the upsampling layer. A last convolution layer is used to obtain a single image followed by a sigmoid layer.

As previously mentioned, the goal of U-Net is to predict the underlying structure of super-resolved images. Training U-Net$_{\text{data map}}$ in a fully supervised manner requires a training dataset of associated super-resolved images and underlying structures. However, such a dataset does not currently exist. Mathematically, a microscopy image is obtained from the convolution of the microscope E-PSF with the position of fluorophores at the sample. In the images from ref. 6, the E-PSF of the microscope can be approximated by a Gaussian function with a full-width at half-maximum of ~50 nm. Hence, U-Net$_{\text{data map}}$ can be trained to predict that the data map that once convolved with the E-PSF will be similar to the input image (Fig. 1f). The L_2 error is calculated between the Gaussian-convolved data map and the original input image as the loss function to minimize.

To train the model, we used good-quality STED images of diverse neuronal proteins from an existing dataset[6] (quality, >0.7). In ref. 6, the quality score of an image was obtained by asking an expert to rate the image based on a qualitative assessment of the resolution of the structure of interests and the signal-to-noise ratio on a scale from 0 to 1. The quality scores from the original dataset were used to train a DL model to automatically rate the quality of an image. Supplementary Table 6 presents the proteins imaged and the number of images that were used for training. Each 224 × 224 pixel^2 image is augmented with three (3) 90° rotations. The Adam optimizer was used with default parameters using a learning rate of 1×10^{-4}. The model was trained for 1,000 epochs with a batch size of 32. We selected the model with the best generalization properties on the validation set, obtained from the mean squared error between the input image and data map after applying Gaussian convolution.

By default, the predicted data map reconstructs the background noise from the image. Filtering can be applied on the predicted data map to reduce the impact of noise. The number of emitters can be adapted to the experimental context, which is then converted into an integer value. U-Net$_{\text{data map}}$ was trained with 224 × 224 pixel^2 images, but images of arbitrary size can be processed at inference time.

Data maps from deconvolution. Data maps were generated using the Richardson–Lucy deconvolution implementation[71]. The E-PSF of the input image was approximated by a Gaussian function with a full-width at half-maximum of ~50 nm. Thirty iterations were used for the deconvolution algorithm.

Imaging optimization objectives

Resolution. We calculated the resolution of the images by using the parameter-free image resolution estimation based on a decorrelation analysis developed in another work[40]. The decorrelation analysis was used due to its simplicity in transferring from simulation to real imaging conditions.

Photobleaching. In all the experiments involving photobleaching as one of the imaging optimization objective, we measured the loss of fluorescence signal between a low-resolution image that is acquired before (Confocal1) and after (Confocal2) the high-resolution (STED) acquisition[6]. Photobleaching is defined as

$$\text{Photobleaching} = \frac{\overline{\text{Confocal1}_{\text{fg}}} - \overline{\text{Confocal2}_{\text{fg}}}}{\overline{\text{Confocal1}_{\text{fg}}}}, \tag{9}$$

where $\overline{\text{ConfocalX}_{\text{fg}}}$ is the average signal on the foreground of the first confocal image (Confocal1). The foreground mask is determined using an Otsu threshold on Confocal1.

Signal ratio. We calculate the signal ratio as the ratio between the intensity in the high-resolution image and the respective confocal image using the following equation:

$$\text{Signal Ratio} = \frac{\text{STED}_{\text{fg}}^{75} - \overline{\text{STED}_{\text{bckg}}}}{\text{Confocal1}_{\text{fg}}^{75}}. \tag{10}$$

The foreground mask of the STED and confocal images is determined using the Otsu method. The foreground signal in the mask is calculated as the 75th percentile of the image (STED or Confocal1). $\overline{\text{STED}_{\text{bckg}}}$ represents the mean signal of the background signal in the STED image.

Artefact. We measured the imaging artefacts with a metric inspired by SQUIRREL[72] and MASK-SSIM[73] approaches. Specifically, we map the super-resolution image to a low-resolution image using a similar procedure to SQUIRREL but compare the structures only within a foreground mask. This foreground mask is obtained using the Otsu method on Confocal1. The average structural similarity index on the foreground between the low-resolution and optimized super-resolution image is reported as the metric. The value of the artefact metric that is reported in this Article is

$$\text{Artefact} = 1 - \text{SSIM}_{\text{fg}}. \tag{11}$$

Comparison of pySTED simulations with real acquisitions

We compared pySTED-simulated images with images acquired on a real STED microscope with similar imaging parameters. To evaluate the reliability of the simulations, we acquired ten images using a different combination of parameters. We varied each imaging parameter over a range that is commonly used for routine STED experiments and that would not damage the microscope (for example detectors, see the parameters listed in Supplementary Table 7). We used a sample of immunostained cultured hippocampal neurons of the neuronal protein bassoon tagged with the fluorophore ATTO-647N. The small clusters formed by bassoon are well suited for measurements of resolution. The same parameter combination is used in pySTED and on the microscope for fair comparison.

We optimized the photobleaching constants (k_1 and b) and the STED cross-section (σ_{STED}) of the fluorophore to match the measured photobleaching and resolution values using a least-squares method (Supplementary Fig. 4a, right). It is implemented iteratively, with the optimization of photobleaching and resolution done sequentially and repeated 15 times, since the optimization of σ_{STED} also affects photobleaching.

For each acquired real STED image, a data map is predicted with U-Net$_{\text{data map}}$. The number of emitters per pixel is obtained by multiplying the data map with a correction factor f to match the fluorescence signal in real images. This correction factor f can be obtained by fitting the intensity value obtained at pixel (x, y) to the real intensity of the acquired confocal image ($I_{\text{CONF}}(x, y)$). The intensity value of the synthetic image is approximated as the product between the E-PSF and data map (D):

$$I_{\text{CONF}}(x, y) = f \sum_{ij} D_{ij}(x, y) \text{E} - \text{PSF}_{ij}. \tag{12}$$

Weakly supervised learning for the segmentation of F-actin nanostructures

We compared three training schemes (five random initializations per scheme) to train a U-Net model to segment two F-actin nanostructures (fibres and rings): (1) original model from ref. 46 (O); (2) a model that uses a quantile normalization of the large image (min–max normalization using the 1st and 99th quantiles) and increased data augmentation during training (N; see below); and (3) a model trained as that in (2) with synthetic images (N + S). In all the conditions, the same architecture, training procedure and dataset are used, as per ref. 46. A model is trained for 700 epochs, and the best model on the validation dataset is kept for testing. To compare only the impact of training, the validation dataset is kept constant in all the training instances.

In all the training schemes, an augmentation has 50% probability of being selected. For the training scheme O (ref. 46), the augmentations consisted of horizontal/vertical flips, intensity scale and gamma adaptation. For the approaches using an increased data augmentation scheme (N and N + S), the augmentations from O are combined with random 90° rotations, crop normalization (1st and 99th percentile) and more intensity scale and gamma adaptation operations.

Synthetic F-actin images. The U-Net$_{\text{data map}}$ model (Fig. 2) was used to extract the data maps of all the valid crops in the training dataset (contains >10% of dendrite, 256 × 256 pixel^2, 25% overlap). Five synthetic images with different resolution and noise properties were simulated for each crop with pySTED using a parameter combination that would minimally allow to resolve the F-actin nanostructures (Fig. 2b and Supplementary Table 8).

Generation of subsets. Models (with constant parameter initialization) were trained on subsets of the original dataset, to evaluate if pySTED can help reduce the number of original images in the training dataset. Five independent subsets with 0.75, 0.5, 0.25, 0.1, 0.05 and 0.025 ratios were used for training. The ratio of 0.025 corresponds to training on a single image (42 images in the training dataset). When an image is discarded from a subset, its corresponding crops (synthetic included) are also removed from training (Supplementary Fig. 7).

Performance evaluation. AP is used for performance evaluation. The AP is obtained from the precision and recall measured by the predicted segmentation compared with the ground-truth manual annotations. The AP corresponds to the area under the $p_*(r)$ curve. $p_*(r)$ is given by the maximum precision value that can be attained at any recall r_i greater than recall r, that is,

$$p_*(r) = \max_{r_i \geq r} p(r_i). \tag{13}$$

The AP is calculated as

$$\mathrm{AP} = \int_0^1 p_*(r)\mathrm{d}r. \tag{14}$$

Multi-objective bandit optimization

The multi-objective bandit optimization aims at finding a set of imaging parameters that simultaneously optimizes all the imaging optimization objectives. Such a multi-objective problem is ill-defined as there exists a set of Pareto optimal objectives that could be used to solve the task. Hence, an external input (for example, a microscopist) is required to make the necessary trade-offs over the course of the optimization session.

Algorithms. The goal of the algorithm is to learn the mapping between imaging parameters (for example, laser power and pixel dwell time) and the imaging optimization objective (for example, resolution, photobleaching, artefact or signal ratio) by exploring the parameter space and exploiting its current knowledge of the parameter space to acquire high-quality images. A single model is built for each optimization objective as that in another work[6]. The exploration/exploitation trade-off is achieved via TS[74]. At each time step of optimization, a function is sampled from the posterior of each model. The expected imaging optimization objective associated with each imaging parameter option is combined. The preferred combination is selected and an image is acquired with the associated parameters. The imaging optimization objectives are calculated from the resulting image and used to update each model.

The range of imaging parameters was normalized in [−1, 1] using a min–max normalization. The min–max values were given from the predefined range of a parameter. All the models are trained from scratch. At the start of each optimization session, three images are acquired with parameter combinations obtained from expert knowledge, allowing the models to gain insights into the imaging task. For further implementations, these parameter combinations could be obtained from (1) a previous imaging session, (2) different fluorophores or (3) publications from the field.

Kernel-TS: it is implemented by following the procedure from another work[6]. The regression model that maps the imaging parameters to the imaging optimization objectives is a non-parametric Gaussian process. All the parameters of the method (for example, kernel bandwidth or bounds on noise) were based on the recommendations from ref. 6. Kernel-TS works on a discrete parameter space of ten points for each optimized parameter. The values of imaging optimization objectives are rescaled using a whitening transformation.

LinTSDiag: it is a neural network implementation of TS[48]. LinTSDiag was previously implemented to solve a two-parameter DyMIN task[49] (Supplementary Table 14). In this implementation, the neural network is a fully connected network with 2 layers with hidden sizes of 32. After each layer, ReLU activation is used and followed by a dropout layer (probability of 0.2). The last layer of the model projects to a single imaging optimization objective value. The model is implemented in PyTorch[70] and relies on the seminal implementation from ref. 48. The loss of the model is the mean squared error and is optimized using stochastic gradient descent, with a learning rate of 1×10^{-3}. After each acquisition, the weights of the model are updated until the error is $<1 \times 10^{-3}$ or 1,000 updates have been done. During training, the imaging optimization objectives are rescaled into a [0, 1] range.

Two parameters (ν and λ (ref. 48)) control the exploration of the model. Increasing their values results in more exploration. In all the experiments using LinTSDiag, $\lambda = 0.1$ is used. The parameter ν varied depending on the task: (1) in simulation, $\nu = 0.01$ (Fig. 3a–c); (2) in four-parameter optimization, $\nu = 0.1$ (Fig. 3e,f); and (3) in six-parameter optimization, $\nu = 0.25$ (Fig. 3g,h).

LinTSDiag handles a continuous parameter space. Hence, it is not possible to display all the possible trade-offs. To reduce the number of possibilities, only the Pareto optimal combination of optimization objectives are displayed (Pareto front). The Pareto optimal options are extracted using NSGA-II (ref. 75) with the implementation from the DEAP Python library[76]. Since computing the Pareto front is computationally expensive, a stopping criterion is used to reduce the calculations[77]. The stopping criterion is based on the rolling standard deviation of the maximum crowding distance (window size of 10) during the NSGA-II search. The search is stopped when the standard deviation is lower than $\sqrt{2 \times 10^{-4} \times n}$, where n is the number of optimization objectives[77]. In theory, the NSGA-II search should restart from scratch after each acquisition. However, given the high dimensionality of the parameter space, this may lead to high variability in the proposed parameter combination. To reduce this variability, a fraction of the previous options was retained as a warm start of the NSGA-II search[78]. In this work, 30% of the previous options are randomly sampled and used as starting points for the next NSGA-II search. The resulting Pareto front of imaging optimization objectives is given as an input to the preference articulation method.

Contextual LinTSDiag: the contextual version of LinTSDiag heavily relies on the implementation of LinTSDiag described above (Supplementary Table 15). In this work, the contextual information was used to solve a DyMIN microscopy task. As previously mentioned, the confocal image serves as contextual information, but any other contextual information pertinent to the task could be provided to the model. The confocal image is encoded with a two-layer CNN. The first convolution layer with 8 filters, kernel size of 3 and padding of 1 is followed by a batch normalization layer, maxpooling layer (size, 4; stride, 4) and ReLU activation. The second convolution layer of 16 filters is followed by a batch normalization layer. Global average pooling is used to generate vector embedding. This is followed by ReLU activation and dropout layer with a probability of 0.2. The embedding is projected to 32 features using a fully connected layer and is followed by ReLU activation and a dropout layer with a probability of 0.2. The contextual features are concatenated with the parameter features (described in LinTSDiag). A single-layer fully connected model with a hidden size of 64 is used to predict the imaging optimization objectives. ReLU activation is used at the hidden layer. A single contextual encoder is created and shared between the imaging optimization objectives. The same training procedure and NSGA-II search are used as in LinTSDiag.

The exploration parameters $\lambda = 0.1$ and $\nu = 0.25$ were used in simulation (Fig. 4c–g) and in three-dimensional DyMIN optimization (Fig. 4h–j).

Preference articulation. The optimization algorithms output possible trade-offs between the imaging optimization objectives. The preference articulation step consists of selecting the trade-off that is the most relevant for the task. Two preference articulation methods were used in bandit optimization: manual selection and automatic selection[6].

Manual selection: this method requests a manual input from the microscopist at each image acquisition. The microscopist is asked to select the trade-off that is in line with their own preferences from the available options (point cloud). This method was used in all the experiments on the real microscope using the bandit optimization scheme (Figs. 3e–h and 4h–j).

Automatic selection: this method aims at reducing the number of interventions from the microscopist in the optimization loop by learning their preferences before the optimization session. In ref. 6, the neural network implementation PrefNet was used to learn the preferences from an expert. In the current work, two PrefNet models were trained from the preferences of an expert. The same model architecture and training procedure were used as that in ref. 6. The first model is trained for STED optimization to select from the imaging optimization objectives including resolution, photobleaching and signal ratio. The second model is trained for the DyMIN optimization to select the trade-off between resolution, photobleaching and artefact. The PrefNet model is used to repeatedly make the trade-offs in multiple optimizations in the simulation environment (Figs. 3b–d and 4c–g).

RL experiments

An RL agent interacts with an environment by sequentially making decisions based on its observations. The goal of the agent is to maximize its reward signal over the course of an episode.

RL formulation: the general problem in RL is formalized by a discrete-time stochastic control process, that is, it satisfies a Markov decision process. An agent starts in a given state $s_t \in \mathbf{S}$ and gathers some partial observations $o_t \in \mathbf{O}$. In a Markov decision process, the state is fully observable, that is, the agent has access to a complete observation of state s_t. At each time step t, the agent performs an action $a_t \in \mathbf{A}$ given some internal policy π after which the agent receives a reward $r_t \in \mathbf{R}$ and transitions to a state $s_{t+1} \in \mathbf{S}$ with a state transition function $\mathbf{T}(s_{t+1}|s_t, a_t)$. Following the state transition, a reward signal $r_t = R(s_t, a_t, s_{t+1})$ is provided to the agent as feedback. The goal of the agent is to maximize the cumulative reward over the trajectory $\tau = (s_t, a_t, s_{t+1}, a_{t+1}...)$. Formally, the cumulative reward may be written in the form of

$$R(\tau) = \sum_{t=0}^{T} \gamma^t r_t, \tag{15}$$

where γ is a discount factor in the range [0, 1] to temporally weight the reward. Intuitively, using a discount factor close to 1 implies that the credit assignment of the current action is important for future reward, which is the case for a long planning horizon, whereas a discount factor close to 0 reduces the impact of temporally distant rewards[50].

Reward function: the optimization of super-resolution STED microscopy is a multi-objective problem (for example, resolution, signal ratio and photobleaching). However, the conventional RL settings and algorithms assume access to a reward function that is single valued, that is, single-objective optimization[50]. Several methods were introduced to solve the multi-objective RL setting, for instance, by simultaneously learning multiple policies or by using a scalarization function (ref. 79 provides a comprehensive review). The scalarization function is simple to implement and allows all the algorithms that were developed for RL to be used, but assumes that the preference from the user is known a priori. In this work, the multi-objective RL setting was transformed into a single scalar reward by using the neural network model PrefNet[6], which was developed in the bandit experiments. Indeed, the PrefNet model was trained to reproduce the trade-off that an expert is willing to make into the imaging optimization objective space. This is done by the PrefNet model by assigning a value to a combination of imaging optimization objectives. The values predicted by the model for a combination of optimization objectives are arbitrary, but the ranking of these values is accurate. Hence, the values from the PrefNet model are proportional to the image quality. The reward of the agent can then be defined using equation (16). For safety precautions, when deploying the agent on a real microscopy system, the agent incurs a reward of −10 when the frequency of photons on the detector is higher than 20 MHz.

$$r_t = \begin{cases} -10 & \text{when} f_{\text{photons}} > 20\,\text{MHz} \\ \text{PrefNet}(R, P, S) & \text{otherwise.} \end{cases} \tag{16}$$

Although a negative reward can be used to limit the selection of actions that could damage the microscope, it is not required. For instance, the results shown in Fig. 6d,e and Extended Data Fig. 4 used a version of the reward function that did not include the negative reward. It is worth noting that in these cases, the range of parameters should be carefully selected to avoid damage to the microscope.

Agent: the PPO model[54] was used for all the RL experiments. PPO is considered to be the state of the art for many control tasks, and is widely used in robotics[51]. PPO allows a continuous action space, making it suitable for the task of microscopy parameter tuning. It is an on-policy algorithm, meaning that the same policy is used during data collection and updating phases. The model uses a deep neural network to map the state to the actions. Since PPO is an actor–critic method, it simultaneously learns a policy function and a value function that measures the quality of a selected action (Supplementary Tables 16 and 17). Both functions use the same model architecture. A convolutional neural network (CNN) extracts information from the visual inputs and a linear neural network (LN) extracts information from the history of the episode. The CNN encoder is similar to the one used in ref. 30. The encoder is composed of three layers of convolutions, each followed by a leaky ReLU activation. The kernel size of each layer is 8, 4 and 3 with a stride of 4, 2 and 1. This allows the spatial size of the state space to be reduced. The LN model contains two linear layers projecting to sizes of 16 and 4. The information from both layers is concatenated and mapped to the action space using an LN layer.

During training, the Adam optimizer is used with default parameters and a learning rate of 1×10^{-4}. The batch size of the model is set at 64. For each of the 512 steps in the environment, the model is trained for 10 batches, which are randomly sampled from the previous 512 steps. A maximal gradient of 1.0 during backpropagation is used to stabilize training.

Synthetic data maps: a bank of data maps was generated using U-Net$_{\text{data map}}$. Supplementary Table 18 presents the number of images per structure that were available during training. Data maps were randomly cropped to 96 × 96 pixel^2 with a higher probability of being sampled within the foreground of the data map. Random data augmentation is performed online with a 50% probability: random [1, 3] 90° rotations, up–down flips and left–right flips. The resultant cropped data map is multiplied by a value that is sampled from $N(\mu = 40, \sigma = 4)$ and turned into an integer array using the floor operation.

Synthetic fluorophores: synthetic fluorophore properties are generated on the fly during training by uniform sampling. Supplementary Table 19 displays the range of possible fluorophore properties. The parameters k_1, b and σ_{abs} are optimized using the procedure described in the 'Comparison of pySTED simulations with real acquisitions' section. During the optimization, it is assumed that the maximum number of emitters is 40. A scaling factor that is dependent on the type of structure is used during the optimization (Supplementary Table 20). At each iteration, the fluorophore parameters from the previous iterations are used as a starting point. The initial conditions of the parameters are $k_1 = 2.9 \times 10^{-16}$, $b = 1.66$ and $\sigma_{\mathrm{abs}} = 3.2 \times 10^{-21}$ m^2.

STED microscopy experiments

STED imaging. Super-resolution imaging of neuronal proteins was performed on an Expert Line STED system (Abberior Instruments) equipped with a ×100 1.4-numerical-aperture oil-objective lens (Olympus, UPLSAPO100XO), motorized stage and auto-focus unit. Far-red dyes were imaged using a 640 nm pulsed diode (40 MHz), a 775 nm depletion laser (40 MHz) and an ET685/70 (Chroma) fluorescence filter. Fluorescence was detected on avalanche photodiode detectors with approximately 1 Airy unit detection pinhole. Images were processed using Fiji (ImageJ, 1.54f) software. One- and two-channel imaging processes of tubulin, NPC and golgi in Vero cells were performed on an Infinity line microscope (Abberior Instruments) using the imaging settings described elsewhere[8].

Before the optimization, the excitation power of the confocal acquisition needed to be set to acquire <200 photons in 10 µs. To do so, the excitation power was first set to 10 µW and was halved until this criterion was met. This value is used by the model to incorporate knowledge about the brightness of the sample.

Kidney epithelial cell culture. Vero B4 cells were obtained from the DSMZ—German Collection of Microorganisms and Cell Cultures. They were maintained in Dulbecco's modified Eagle's medium (Gibco) supplemented with GlutaMAX (Thermo Fisher Scientific), 10% foetal bovine serum (Sigma-Aldrich), 1 mM sodium pyruvate (Thermo Fisher Scientific) and penicillin–streptomycin (100 µl ml^{-1} and 0.1 mg ml^{-1}; Sigma-Aldrich) at 37 °C with 5% CO_2.

Sample preparation and staining procedures: for indirect immunostaining, cells were fixed in 8% paraformaldehyde in phosphate-buffered saline (PBS) and permeabilized with 0.5% Triton X-100/PBS for NPC proteins (Mab414, 1:200; abcam, code ab24609) and golgi (Giantin, 1:200; abcam, code ab80864) staining. Methanol was used as a fixative for tubulin staining (1:500; abcam, code ab18251).

After blocking with 2% bovine serum albumin/0.1% TWEEN 20/PBS, cells were incubated with the primary antibody for 1 h at the specified dilutions. Detection of primary antibodies was achieved using secondary STAR RED goat anti-mouse IgG (1:200, abberior, code STRED-1001-500UG) and STAR ORANGE goat anti-rabbit IgG (1:200, abberior, code STORANGE-1002-500UG) for the double staining of NPC and golgi. Tubulin was labelled with STAR RED goat anti-rabbit IgG (1:200, abberior, code STRED-1002-500UG). Secondary antibodies were also incubated for 1 h.

After stringent washing with PBS, cells were mounted in MOUNT SOLID ANTIFADE (abberior, code MM-2013-2X15ML). The protocol was adapted from ref. 80.

Neuronal cell culture. Neuronal cultures from the hippocampus were obtained using neonatal Sprague Dawley rats. The rats, aged P0–P1, were sacrificed through decapitation before the hippocampi were dissected following procedures approved by the Animal Care Committee of Université Laval. The cells were then seeded onto 12 and 18 mm coverslips coated with poly-D-lysine and laminin, for fixed (12 mm, 40,000 per coverslip) and live-cell (18 mm, 100,000 cells per coverslip) STED imaging. Neurons were cultivated in a growth medium composed of Neurobasal and B27 (in a 50:1 ratio), enriched with penicillin–streptomycin (25 U ml^{-1}/25 µg ml^{-1}) and 0.5 mM L-GlutaMAX (Invitrogen). Ara-C (5 µM; Sigma-Aldrich) was added into the medium after five days to limit the proliferation of non-neuronal cells. Twice a week, ~50% of the growth medium was replaced with serum- and Ara-C-free medium. Cells were used between 12 and 16 days for in vitro experiments.

Sample preparation and staining procedures: fixation was performed for 10 min in 4% paraformaldehyde solution (paraformaldehyde, 4%; sucrose, 4%; phosphate buffer, 100 mM; Na-EGTA, 2 mM). Neurons were permeabilized with 0.1% Triton X-100 and aspecific binding sites were blocked for 30 min with 2% goat serum in 20 mM PBS. Primary and secondary antibodies were successively incubated for 2 and 1 h, respectively. Phalloidin was incubated for 1 h. All the incubations were done at room temperature in the blocking solution. Immunostained coverslips were mounted in Mowiol-DABCO for imaging. F-actin was stained with phalloidin-STAR635 (Sigma-Aldrich, cat. no. 30972-20µg, 1:50 dilution). All the antibodies used in this study with the associated concentrations are listed in Supplementary Tables 21 and 22.

For the live-cell experiment (Fig. 6d), the neurons were incubated for 8 min with SiR-actin (0.8 µM, cytoskeleton, cat. no. CY-SC001) diluted in HEPES-buffered artificial cerebrospinal fluid (NaCl, 98; KCl, 5; HEPES, 10; glucose, 10; $CaCl_2$, 0.6; $MgCl_2$, 5; all values are in mM). For live-cell STED microscopy, coverslips were mounted on a quick-release chamber (Wagner Instruments, cat. no. 61-1944) and imaged in HEPES-buffered artificial cerebrospinal fluid comprising 5 mM Mg^{2+}/0.6 mM Ca^{2+} using a gravity perfusion system.

Quantification of biological structures. F-actin: line profiles of ~1 µm were manually extracted from each image. A linewidth of 3 pixels was used to average the profile values. The autocorrelation function (statsmodels library[81]) was calculated from the intensity profile. The length of periodicity of the signal was determined from the first peak maximum.

CaMKII-β and PSD95: we segmented the clusters using wavelet segmentation from a previous implementation[82]. The scales used were (1, 2) for STED and (3, 4) for confocal segmentation. A threshold of 200 was used. Small segmentation objects (<3 pixels) were removed and small holes (<6 pixels) were filled. In the STED image segmentation, only the object part of the confocal foreground was considered. For STED segmentation, watershed was used to split the merged segmented objects. The local peak maximum was used as the initial seed. Small segmentation objects (<3 pixels) resulting from the watershed split were filtered out. The properties of each segmented object was extracted using regionprops from the scikit-image Python library[71].

TOM20: a similar approach to the one in ref. 60 was used. Briefly, the confocal foreground of each mitochondrion was extracted using the same wavelet segmentation procedure as that used for CaMKII-β and PSD95. The two-dimensional autocorrelation on square crops of 320 nm × 320 nm centred on each mitochondrion was calculated. The diameter of the TOM20 cluster is defined as the standard deviation obtained from a two-dimensional Gaussian curve fit of the autocorrelation profile.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

All the datasets used to train the models and the corresponding acquired images are available for download at https://s3.valeria.science/flclab-pysted/index.html. Source data are provided with this paper.

Code availability

All code used in this paper is open source. Code for the STED simulation platform, pySTED, is available via GitHub at https://github.com/FLClab/pySTED (ref. 83). The bandit algorithms and training procedures are available via GitHub at https://github.com/FLClab/optim-sted (ref. 84). The gym-STED environment and training routines are available via GitHub at https://github.com/FLClab/gym-sted (ref. 85) and https://github.com/FLClab/gym-sted-pfrl (ref. 86), respectively.

References

1. Schermelleh, L. et al. Super-resolution microscopy demystified. *Nat. Cell Biol.* **21**, 72–84 (2019).
2. Hell, S. W. & Wichmann, J. Breaking the diffraction resolution limit by stimulated emission: stimulated-emission-depletion fluorescence microscopy. *Opt. Lett.* **19**, 780–782 (1994).
3. Oracz, J., Westphal, V., Radzewicz, C., Sahl, S. J. & Hell, S. W. Photobleaching in STED nanoscopy and its dependence on the photon flux applied for reversible silencing of the fluorophore. *Sci. Rep.* **7**, 11354 (2017).
4. Tosheva, K. L., Yuan, Y., Pereira, P. M., Culley, S. & Henriques, R. Between life and death: strategies to reduce phototoxicity in super-resolution microscopy. *J. Phys. D Appl. Phys.* **53**, 163001 (2020).
5. Boudreau, C. et al. Excitation light dose engineering to reduce photo-bleaching and photo-toxicity. *Sci. Rep.* **6**, 30892 (2016).
6. Durand, A. et al. A machine learning approach for online automated optimization of super-resolution optical microscopy. *Nat. Commun.* **9**, 5247 (2018).
7. Dreier, J. et al. Smart scanning for low-illumination and fast RESOLFT nanoscopy in vivo. *Nat. Commun.* **10**, 556 (2019).
8. Heine, J. et al. Adaptive-illumination STED nanoscopy. *Proc. Natl Acad. Sci. USA***114**, 9797–9802 (2017).
9. Vinçon, B., Geisler, C., Egner, A. & Egner, A. Pixel hopping enables fast STED nanoscopy at low light dose. *Opt. Express* **28**, 4516–4528 (2020).
10. Scherf, N. & Huisken, J. The smart and gentle microscope. *Nat. Biotechnol.* **33**, 815–818 (2015).
11. Krull, A., Hirsch, P., Rother, C., Schiffrin, A. & Krull, C. Artificial-intelligence-driven scanning probe microscopy. *Commun. Phys.* **3**, 54 (2020).
12. Moen, E. et al. Deep learning for cellular image analysis. *Nat. Methods* **16**, 1233–1246 (2019).
13. von Chamier, L. et al. Democratising deep learning for microscopy with ZeroCostDL4Mic. *Nat. Commun.* **12**, 2276 (2021).
14. Maška, M. et al. The Cell Tracking Challenge: 10 years of objective benchmarking. *Nat. Methods* **20**, 1010–1020 (2023).
15. Ljosa, V., Sokolnicki, K. L. & Carpenter, A. E. Annotated high-throughput microscopy image sets for validation. *Nat. Methods* **9**, 637–637 (2012).
16. Vinyals, O. et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* **575**, 350–354 (2019).
17. Levine, S., Pastor, P., Krizhevsky, A. & Quillen, D. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *Int. J. Rob. Res.* **37**, 421–436 (2018).
18. Chaware, A., Cooke, C. L., Kim, K. & Horstmeyer, R. Towards an intelligent microscope: adaptively learned illumination for optimal sample classification. In *ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 9284–9288 (IEEE, 2020).
19. Turcotte, B., Bilodeau, A., Lavoie-Cardinal, F. & Durand, A. pySTED: a STED microscopy simulation tool for machine learning training. In *Association for the Advancement of Artificial Intelligence—AAAI Workshop on AI to Accelerate Science and Engineering* (2022).
20. Kalinin, S. V. et al. Probe microscopy is all you need. *Mach. Learn.: Sci. Technol.* **4**, 023001 (2023).
21. Adhikari, M., Houhou, R., Hniopek, J. & Bocklitz, T. Review of fluorescence lifetime imaging microscopy (FLIM) data analysis using machine learning. *J. Exp. Theor. Anal.* **1**, 44–63 (2023).
22. Sage, D. et al. Super-resolution fight club: assessment of 2D and 3D single-molecule localization microscopy software. *Nat. Methods* **16**, 387–395 (2019).
23. Lagardère, M., Chamma, I., Bouilhol, E., Nikolski, M. & Thoumine, O. FluoSim: simulator of single molecule dynamics for fluorescence live-cell and super-resolution imaging of membrane proteins. *Sci. Rep.* **10**, 19954 (2020).
24. Novák, T., Gajdos, T., Sinkó, J., Szabó, G. & Erdélyi, M. TestSTORM: versatile simulator software for multimodal super-resolution localization fluorescence microscopy. *Sci. Rep.* **7**, 951 (2017).
25. Nehme, E., Weiss, L. E., Michaeli, T. & Shechtman, Y. Deep-STORM: super-resolution single-molecule microscopy by deep learning. *Optica* **5**, 458–464 (2018).
26. Xie, H., Liu, Y., Jin, D., Santangelo, P. J. & Xi, P. Analytical description of high-aperture STED resolution with 0-2π vortex phase modulation. *J. Opt. Soc. Am. A* **30**, 1640–1645 (2013).
27. Caprile, F., Masullo, L. A. & Stefani, F. D. PyFocus—a Python package for vectorial calculations of focused optical fields under realistic conditions. Application to toroidal foci. *Comput. Phys. Commun.* **275**, 108315 (2022).
28. Sarmento, M. J. et al. Exploiting the tunability of stimulated emission depletion microscopy for super-resolution imaging of nuclear structures. *Nat. Commun.* **9**, 3415 (2018).
29. Leutenegger, M., Eggeling, C. & Hell, S. W. Analytical description of STED microscopy performance. *Opt. Express* **18**, 26417–26429 (2010).
30. Mnih, V. et al. Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).
31. Van Rossum, G. & Drake, F. L. *Python 3 Reference Manual* (Create Space, 2009).
32. Sekh, A. A. et al. Physics-based machine learning for subcellular segmentation in living cells. *Nat. Mach. Intell.* **3**, 1071–1080 (2021).
33. Bolaños, L. A. et al. A three-dimensional virtual mouse generates synthetic training data for behavioral analysis. *Nat. Methods* **18**, 378–381 (2021).
34. Zhang, Y. et al. Rapid detection of neurons in widefield calcium imaging datasets after training with synthetic data. *Nat. Methods* **20**, 747–754 (2023).
35. Ronneberger, O., Fischer, P. & Brox, T. U-Net: convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention - MICCAI* **9351**, 234–241 (2015).
36. Falk, T. et al. U-Net: deep learning for cell counting, detection, and morphometry. *Nat. Methods* **16**, 67–70 (2019).
37. He, Y. et al. Surpassing the resolution limitation of structured illumination microscopy by an untrained neural network. *Biomed. Opt. Express* **14**, 106–117 (2023).
38. Qiao, C. et al. Rationalized deep learning super-resolution microscopy for sustained live imaging of rapid subcellular processes. *Nat. Biotechnol.* **41**, 367–377 (2022).
39. ATTO 647N; https://www.atto-tec.com/ATTO-647N.html?language=en
40. Descloux, A., Grußmayer, K. S. & Radenovic, A. Parameter-free image resolution estimation based on decorrelation analysis. *Nat. Methods* **16**, 918–924 (2019).
41. Drobizhev, M., Makarov, N. S., Tillo, S. E., Hughes, T. E. & Rebane, A. Two-photon absorption properties of fluorescent proteins. *Nat. Methods* **8**, 393–399 (2011).
42. Faklaris, O. et al. Quality assessment in light microscopy for routine use through simple tools and robust metrics. *J. Cell Biol.* **221**, e202107093 (2022).
43. Jahr, W., Velicky, P. & Danzl, J. G. Strategies to maximize performance in stimulated emission depletion (STED) nanoscopy of biological specimens. *Methods* **174**, 27–41 (2019).
44. Stringer, C., Wang, T., Michaelos, M. & Pachitariu, M. Cellpose: a generalist algorithm for cellular segmentation. *Nat. Methods* **18**, 100–106 (2021).
45. Ouyang, W. et al. BioImage model zoo: a community-driven resource for accessible deep learning in bioImage analysis. Preprint at *bioRxiv* https://doi.org/10.1101/2022.06.07.495102 (2022).

46. Lavoie-Cardinal, F. et al. Neuronal activity remodels the F-actin based submembrane lattice in dendrites but not axons of hippocampal neurons. *Sci. Rep.* **10**, 11960 (2020).
47. Lamiable, A. et al. Revealing invisible cell phenotypes with conditional generative modeling. *Nat. Commun.* **14**, 6386 (2023).
48. Zhang, W., Zhou, D., Li, L. & Gu, Q. Neural Thompson sampling. *International Conference on Representation Learning* https://doi.org/10.48550/arXiv.2010.00827 (2021).
49. Bilodeau, A., Bernatchez, R., Michaud-Gagnon, A., Lavoie-Cardinal, F. & Durand, A. Contextual bandit optimization of super-resolution microscopy. In *Proc. Canadian Conference on Artificial Intelligence* https://doi.org/10.21428/594757db.9d610b29 (2022).
50. Sutton, R. S. & Barto, A. G. *Reinforcement Learning: An Introduction* 2nd edn (The MIT Press, 2015).
51. Zhao, W., Queralta, J. P. & Westerlund, T. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)* 737–744 (IEEE, 2020).
52. Miao, Q., Lv, Y., Huang, M., Wang, X. & Wang, F.-Y. Parallel learning: overview and perspective for computational learning across Syn2Real and Sim2Real. *IEEE CAA J. Autom. Sin.* **10**, 603–631 (2023).
53. Brockman, G. et al. OpenAI Gym. Preprint at https://arxiv.org/abs/1606.01540 (2016).
54. Schulman, J., Wolski, F., Dhariwal, P., Radford, A. & Klimov, O. Proximal policy optimization algorithms. Preprint at https://arxiv.org/abs/1707.06347 (2017).
55. Risi, S. & Togelius, J. Increasing generality in machine learning through procedural content generation. *Nat. Mach. Intell.* **2**, 428–436 (2020).
56. Xu, K., Zhong, G. & Zhuang, X. Actin, spectrin, and associated proteins form a periodic cytoskeletal structure in axons. *Science* **339**, 452–456 (2013).
57. D'Este, E., Kamin, D., Göttfert, F., El-Hady, A. & Hell, S. W. STED nanoscopy reveals the ubiquity of subcortical cytoskeleton periodicity in living neurons. *Cell Rep.* **10**, 1246–1251 (2015).
58. Ferreira, J. S. et al. Distance-dependent regulation of NMDAR nanoscale organization along hippocampal neuron dendrites. *Proc. Natl Acad. Sci. USA* **117**, 24526–24533 (2020).
59. Nair, D. et al. Super-resolution imaging reveals that AMPA receptors inside synapses are dynamically organized in nanodomains regulated by PSD95. *J. Neurosci.* **33**, 13204–13224 (2013).
60. Wurm, C. A. et al. Nanoscale distribution of mitochondrial import receptor Tom20 is adjusted to cellular conditions and exhibits an inner-cellular gradient. *Proc. Natl Acad. Sci. USA* **108**, 13546–13551 (2011).
61. Staudt, T. M. *Strategies to Reduce Photobleaching, Dark State Transitions and Phototoxicity in Subdiffraction Optical Microscopy*. PhD thesis, Univ. Heidelberg (2009).
62. Shroff, H., Testa, I., Jug, F. & Manley, S. Live-cell imaging powered by computation. *Nat. Rev. Mol. Cell Biol.* **25**, 443–463 (2024).
63. Bouchard, C. et al. Resolution enhancement with a task-assisted GAN to guide optical nanoscopy image analysis and acquisition. *Nat. Mach. Intell.* **5**, 830–844 (2023).
64. Alvelid, J., Damenti, M., Sgattoni, C. & Testa, I. Event-triggered STED imaging. *Nat. Methods* **19**, 1268–1275 (2022).
65. Christiano, P. F. et al. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems* **30** (Curran Associates, 2017).
66. Rafailov, R. et al. Direct preference optimization: your language model is secretly a reward model. *Adv. Neural Inf. Process. Syst.* **36**, 53728–53741 (2023).
67. Willig, K. I., Keller, J., Bossi, M. & Hell, S. W. STED microscopy resolves nanoparticle assemblies. *New J. Phys.* **8**, 106–106 (2006).
68. Hoeller, M. *Advanced Fluorescence Fluctuation Spectroscopy with Pulsed Interleaved Excitation*. PhD thesis, Ludwig-Maximilians-Univ. München (2011).
69. Vicidomini, G. et al. STED nanoscopy with time-gated detection: theoretical and experimental aspects. *PLoS ONE* **8**, e54421 (2013).
70. Paszke, A. et al. PyTorch: an imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems* **32** (Curran Associates, 2019).
71. van der Walt, S. et al. Scikit-image: image processing in Python. *PeerJ* **2**, e453 (2014).
72. Culley, S. et al. Quantitative mapping and minimization of super-resolution optical imaging artifacts. *Nat. Methods* **15**, 263–266 (2018).
73. Gao, S. et al. DETECTOR: structural information guided artifact detection for super-resolution fluorescence microscopy image. *Biomed. Opt. Express* **12**, 5751–5769 (2021).
74. Thompson, W. R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* **25**, 285–294 (1933).
75. Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**, 182–197 (2002).
76. Fortin, F.-A., Rainville, F.-M. D., Gardner, M.-A., Parizeau, M. & Gagné, C. DEAP: evolutionary algorithms made easy. *J. Mach. Learn. Res.* **13**, 2171–2175 (2012).
77. Roudenko, O. & Schoenauer, M. A steady performance stopping criterion for Pareto-based evolutionary algorithms. In *6th International Multi-Objective Programming and Goal Programming Conference* (2004).
78. Deb, K., Rao N., U. B. & Karthik, S. Dynamic multi-objective optimization and decision-making using modified NSGA-II: a case study on hydro-thermal power scheduling. in *Evolutionary Multi-Criterion Optimization* Vol. 4403 (Springer, 2007).
79. Hayes, C. F. et al. A practical guide to multi-objective reinforcement learning and planning. *Auton. Agents Multi-Agent Syst.* **36**, 26 (2022).
80. Wurm, C. A., Neumann, D., Schmidt, R., Egner, A. & Jakobs, S. Sample preparation for STED microscopy. in *Live Cell Imaging: Methods and Protocols* Vol. 591 (Humana Press, 2010).
81. Seabold, S. & Perktold, J. Statsmodels: econometric and statistical modeling with Python. In *Proc. Python in Science Conference* 92–96 (2010). https://doi.org/10.25080/Majora-92bf1922-011 (2010).
82. Wiesner, T. et al. Activity-dependent remodelling of synaptic protein organization revealed by high throughput analysis of STED nanoscopy images. *Front. Neural Circuits* **14**, 57 (2020).
83. Bilodeau, A., Turcotte, B. & Michaud-Gagnon, A. FLClab/pySTED: New release! v0.0.2. *Zenodo* https://doi.org/10.5281/zenodo.13174019 (2024).
84. Bilodeau, A. FLClab/optim-sted: First release! *Zenodo* https://doi.org/10.5281/zenodo.13175942 (2024).
85. Bilodeau, A. & Turcotte, B. FLClab/gym-sted: First release! v0.1.1. *Zenodo* https://doi.org/10.5281/zenodo.13174021 (2024).
86. Bilodeau, A. & Turcotte, B. FLClab/gym-sted-pfrl: New release! v0.1.0. https://doi.org/10.5281/zenodo.13174024 (2024).

Acknowledgements

We thank S. Pensivy and T. Roy from the Neuronal Cell Culture Platform of the CERVO Brain Research Center for the preparation of the dissociated hippocampal cultures, V. Clavet-Fournier for immunostainings of neuronal proteins and M. Bilodeau for help with the pySTED logo. Funding was provided by grants from the Natural Sciences and Engineering Research Council of Canada (NSERC)

(RGPIN-06704-2019 to F.L.-C.), Fonds de Recherche Nature et Technologie (FRQNT) Team Grant (2021-PR-284335 to F.L.-C. and A.D.), the Canadian Institute for Health Research (CIHR) (F.L.-C.) and the Neuronex Initiative (National Science Foundation 2014862, Fonds de recherche du Québec–Santé 295824) (F.L.-C.). A.D. is a CIFAR Canada AI Chair, and F.L.-C. is a Canada Research Chair Tier II. A.B. was supported by a scholarship from NSERC. A.B. and A.M.-G. were awarded an excellence scholarship from the FRQNT strategic cluster UNIQUE.

Author contributions

A.B., A.D. and F.L.-C. designed the study. A.B., A.M.-G., B.T., A.D. and F.L.-C. developed the pySTED simulation platform. A.B. performed the DL and RL experiments and evaluated their performance. A.B. and A.M.-G. evaluated the performance of the pySTED platform and designed the ML experiments. A.B., J.C. and J.H. performed the live-cell and two-colour experiments. A.B. evaluated the deployment of ML and RL strategies in real experimental settings. A.B. and F.L.-C. wrote the paper.

Competing interests

CoLaboratory is a trademark of Google LLC and this paper is not endorsed by or affiliated with Google in anyway. J.H. is an employee of Abberior Instruments—a company manufacturing STED microscopes.

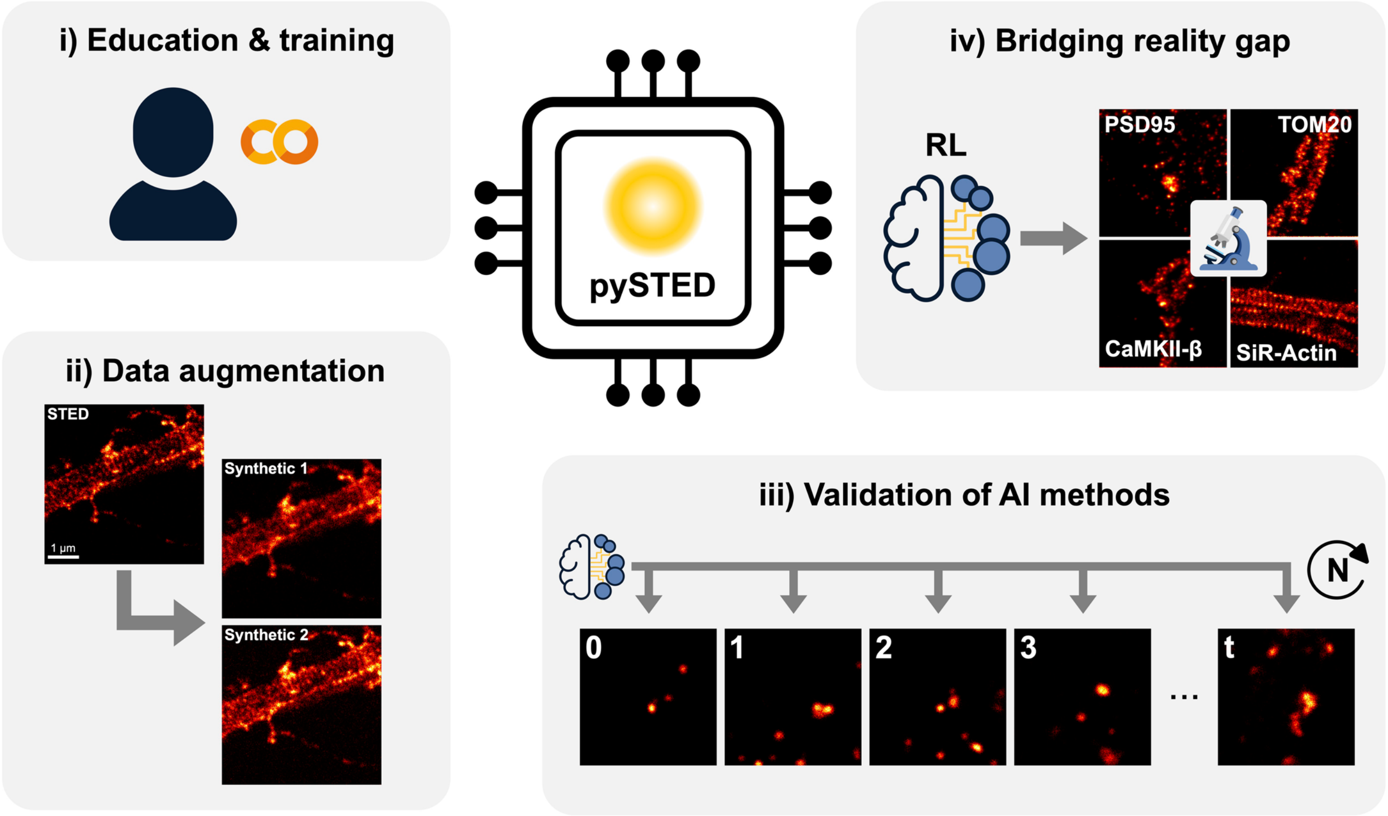
Additional information

Extended data is available for this paper at https://doi.org/10.1038/s42256-024-00903-w.

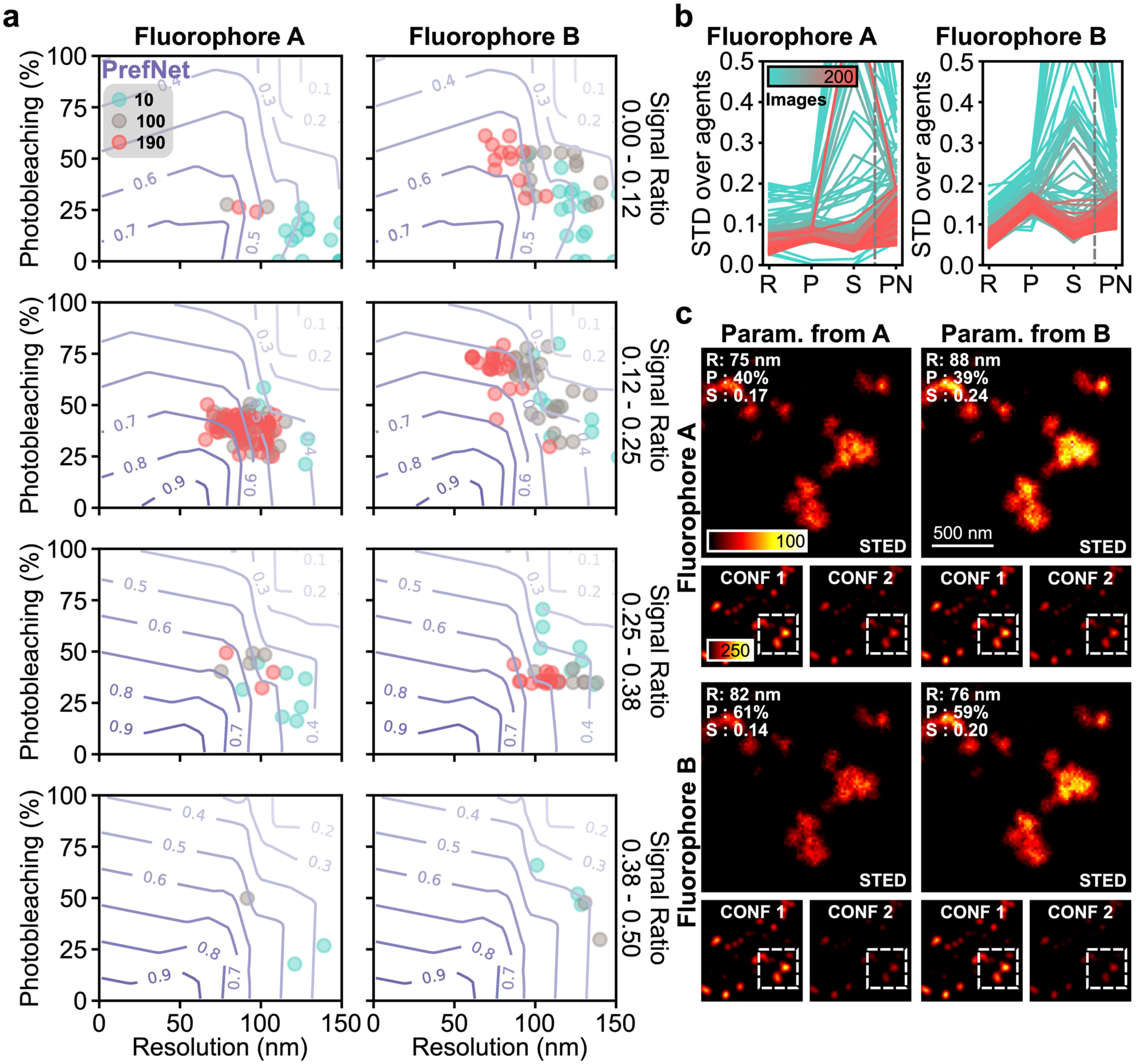
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-024-00903-w.

Correspondence and requests for materials should be addressed to Flavie Lavoie-Cardinal.

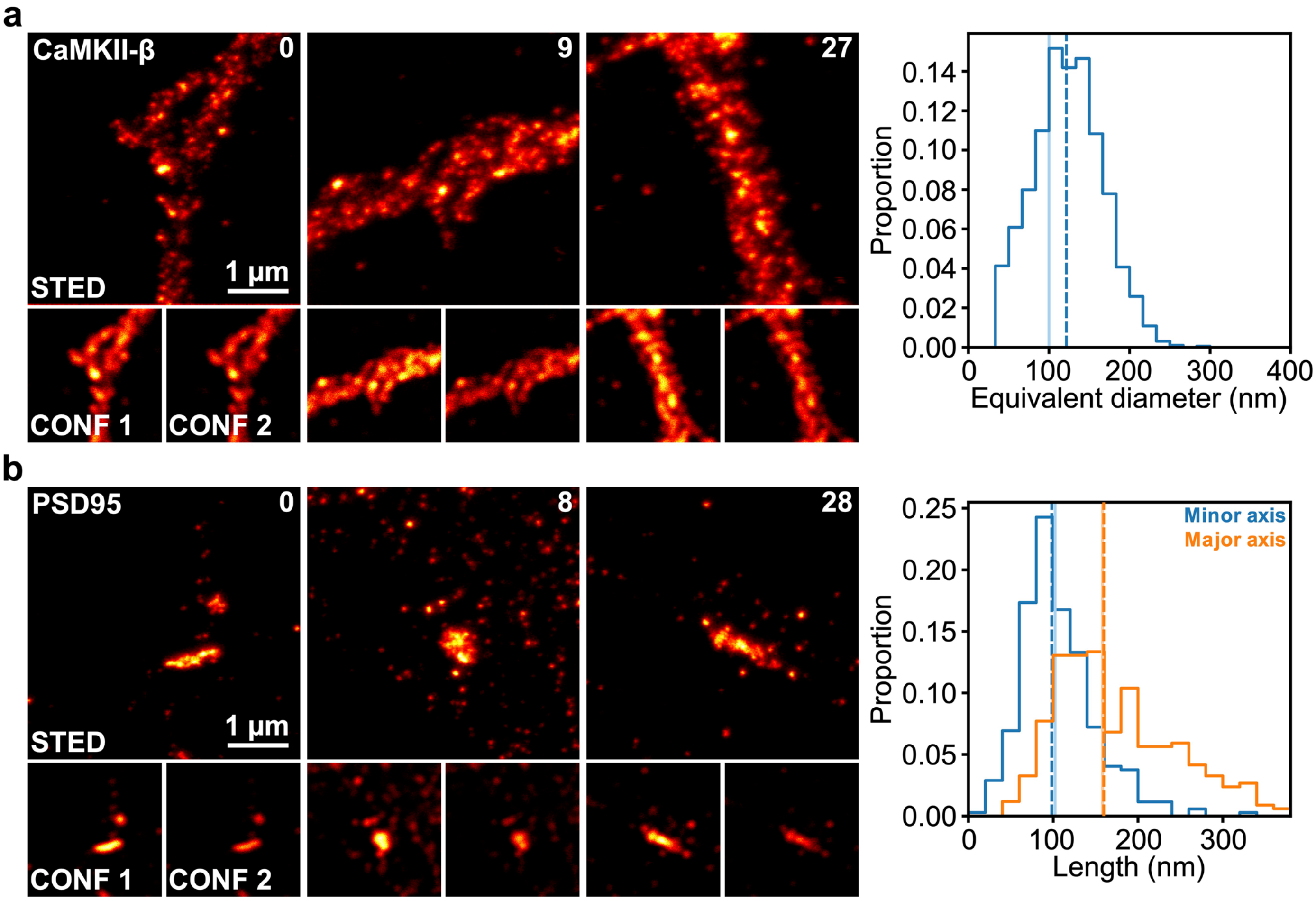
Peer review information *Nature Machine Intelligence* thanks Pierre Jouchet, Hector Santos and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.



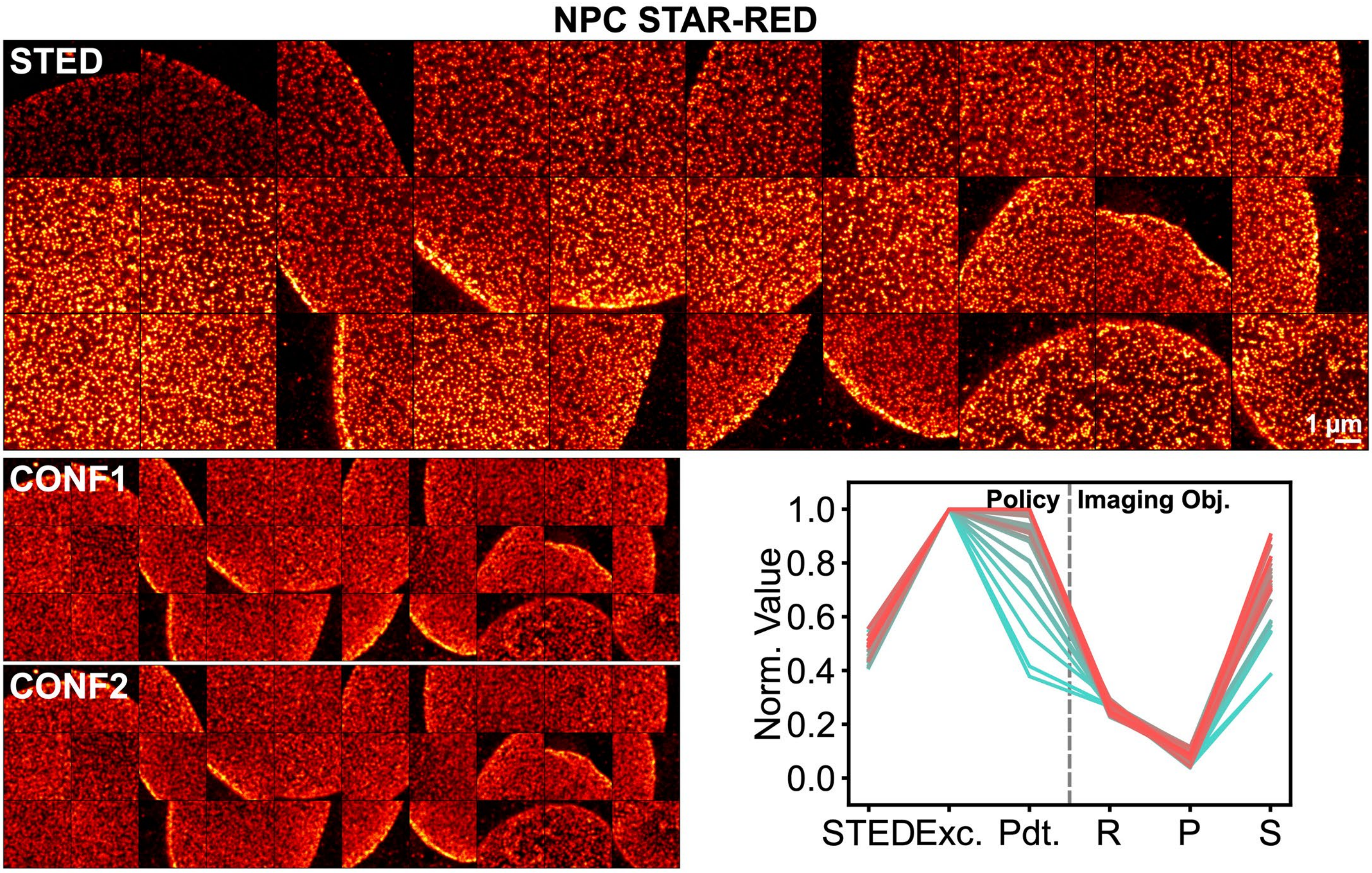
Extended Data Fig. 1 | pySTED can benefit common microscopy tasks, that is image analysis and acquisition. i) A CoLaboratory notebook implementing pySTED is created for trainees to develop their knowledge and intuition about STED microscopy. ii) pySTED can be leveraged in DL-related microscopy tasks to artificially augment the training datasets. iii) pySTED can be used to develop and thoroughly validate AI methods by limiting the impact of biological variability on the measurements and reducing the biological footprint. iv) pySTED reduces the reality gap between simulation and reality by training RL models that learn through interactions with the system. The trained models are then deployed in a wide range of real experimental conditions.



Extended Data Fig. 2 | Optimization of imaging parameters using Kernel-TS. a) Resulting imaging optimization objectives from Kernel-TS at 3 different timesteps (10 - cyan, 100 - grey, and 190 - red) for 50 independent models are presented for increasing signal ratio (top to bottom). With time, Kernel-TS acquires images that have a higher preference score for both fluorophores (purple contour lines) and converges into a similar imaging optimization objective space (red points). **b**) The standard deviation (STD) of the imaging optimization objectives and of the preference scores decreases during the optimization (cyan to red) supporting the convergence of Kernel-TS in a specific region of the imaging optimization objective space for both fluorophores. The dashed line separates the imaging optimization objectives (R: Resolution, P: Photobleaching, and S: Signal ratio) from the preference network (PN). **c**) Typical pySTED simulations on two different fluorophores (top/bottom) using the optimized parameters on fluorophore A (top) or B (bottom). See Supplementary Tab. 10 for imaging parameters.



Extended Data Fig. 3 | The agent was deployed on a real STED microscope for the imaging of diverse neuronal proteins in dissociated hippocampal cultures. In distribution a) (left) The clusters of CaMKII-β are revealed in all acquired images. (right) The size of the clusters is extracted from the acquired images (dashed vertical line represents the median of the distribution) and compared to the values that were previously reported in[58] (solid vertical line). **Out of distribution b**) A bright fluorophore associated to PSD95 is simulated experimentally (Methods). The STED images reveal the presence of nano-clusters. The minor (blue) and major (orange) axis length of the nano-clusters are measured and compared with the values reported in[59].



Extended Data Fig. 4 | Images acquired by the RL agent in a real experiment on a different microscope. NPC was stained with the STAR-RED fluorophore. The sequence of acquired images goes from top left to bottom right. The confocal images before (CONF1) and after (CONF2) are presented for photobleaching comparison. The CONF1 image is normalized to the CONF2 image. The STED images are normalized to the 99th percentile of the intensity of the CONF1 image. Images are 5.12μm × 5.12μm. The evolution of the parameter selection (left; STED: STED power, Exc.: Excitation power, Pdt.: Pixel dwelltime) and imaging optimization objectives (right; R: Resolution, P: Photobleaching, S: Signal ratio) are presented.

nature portfolio

Corresponding author(s): Flavie Lavoie-Cardinal

Last updated by author(s): Aug 8, 2024

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
□	☒	The exact sample size (n) for each experimental group/condition, given as a discrete number and unit of measurement
□	☒	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
□	☒	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	□	A description of all covariates tested
☒	□	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
□	☒	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
□	☒	For null hypothesis testing, the test statistic (e.g. F, t, r) with confidence intervals, effect sizes, degrees of freedom and P value noted *Give P values as exact values whenever suitable.*
☒	□	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	□	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
□	☒	Estimates of effect sizes (e.g. Cohen's d, Pearson's r), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection

Synthetic data experiments were all conducted in pySTED. All codes used in this manuscript are open source. The following libraries were developed and used to collect synthetic data. Code for the STED simulation platform, pySTED is available from https://github.com/FLClab/pySTED. The bandit algorithms and training procedures are available from https://github.com/FLClab/optim-sted. The gym-sted environment and training routines are available from https://github.com/FLClab/gym-sted and https://github.com/FLClab/gym-sted-pfrl respectively. Experiments on a real microscope were acquired with the provided code from optim-sted and gym-sted-pfrl. The code interacted with the Imspector software (16.3.15513) from the Abberior STED microscope. Super-resolution imaging of neuronal proteins was performed on an Abberior Expert-Line STED system (Abberior Instruments GmbH, Germany) equipped with a 100x 1.4NA, oil objective (Olympus, UPLSAPO100XO), motorized stage and auto-focus unit. Far-red dyes were imaged using a 640 nm pulsed diode (40 MHz), a 775 nm depletion laser (40 MHz) and a ET685/70 (Chroma, USA) fluorescence filter. Fluorescence was detected on an avalanche photodiode detectors (APD) with approximately 1 Airy unit detection pinhole. Images were processed using FIJI (ImageJ, 1.54f) software. Single- and two-channel imaging of tubulin, NPC, and Golgi in Vero cells was performed on an Infinity line microscope (Abberior Instruments GmbH, Germany) using imaging settings as described in Heine, et al., (2017) PNAS.

Python libraries used for data collection
DEAP: 1.4.1
gym: 0.26.2
Matplotlib: 3.7.5
Numpy: 1.23.5
Pandas: 2.0.3
Scikit-Image: 0.24.0
Scipy: 1.10.1

Statsmodels: 0.14.2
Tifffile: 2023.7.10
Torch: 1.13.1

Data analysis

All the analysis routines were custom-written in Python (3.10). The analysis routines are available within the GitHub repositories provided above using an open-source license.

Python libraries used for data analysis:
DEAP: 1.4.1
gym: 0.26.2
Matplotlib: 3.7.5
Numpy: 1.23.5
Pandas: 2.0.3
Scikit-Image: 0.24.0
Scipy: 1.10.1
Statsmodels: 0.14.2
Tifffile: 2023.7.10
Torch: 1.13.1

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

All the datasets used to train the models and the corresponding acquired images are available for download at https://s3.valeria.science/flclab-pysted/index.html.

Research involving human participants, their data, or biological material

Policy information about studies with human participants or human data. See also policy information about sex, gender (identity/presentation), and sexual orientation and race, ethnicity and racism.

Reporting on sex and gender

N/A

Reporting on race, ethnicity, or other socially relevant groupings

N/A

Population characteristics

N/A

Recruitment

N/A

Ethics oversight

N/A

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size

For the training of the deep learning models, all images were kept for training. The F-actin training dataset was split into 70%/20%/10% (training/validation/testing) and the training of the U-Net for datamap generation was 70%/20%/10%.
For the optimization experiments, 200 time steps were chosen based on preliminary experiments evaluating the convergence of the models on the microscopy optimization tasks. We measured that the majority of the models can converge to a solution within 200 steps, which was necessary to measure the optimal parameter combinations for different experimental conditions. However, as determined in a previous publication (Durand et al, 2018, Nature Communications), there is an important time limitation for experiments on real samples, especially in

living cells. We chose to evaluate the model after 30 steps, which represents less than 15 minutes (around 30 seconds per step, dependent on the parameter space for optimization), which would be realistic in real experimental settings.

Data exclusions

We did not exclude any data from the analysis.

Replication

Where applicable, we repeated the training of the models using different random seeds to validate the reproducibility of the model.
For bandit optimization in simulation, each experiments was repeated 50 times. All attempts are shown in the manuscript.
For bandit optimization on real samples, experiments were repeated on different samples and therefore were not averaged.
For the RL experiments in simulation, a single model was trained. The model was evaluated on simulated experiments (30 repetitions) on different fluorophores.
For the RL experiments on real experiments, the model that was trained in simulation was evaluated on 7 independant samples. All attempts were successful.

Randomization

We randomly allocated the samples through all stages of the experiments.

Blinding

Blinding was not used in the study as we did not conduct experiments required user bias.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study	
□	☒	Antibodies
□	☒	Eukaryotic cell lines
☒	□	Palaeontology and archaeology
□	☒	Animals and other organisms
☒	□	Clinical data
☒	□	Dual use research of concern
☒	□	Plants

Methods

n/a	Involved in the study	
☒	□	ChIP-seq
☒	□	Flow cytometry
☒	□	MRI-based neuroimaging

Antibodies

Antibodies used

Neuronal cell culture
Primary and secondary antibodies were successively incubated for 2h and 1h respectively. Phalloidin was incubated for 1h. All incubations were done at room temperature, in the blocking solution. Immunostained coverslips were mounted in Mowiol-DABCO for imaging. F-actin was stained with phalloidin-STAR635 (Sigma Aldrich, cat. 30972-20µg, 1:50 dilution). All antibodies used in this study with associated concentrations are provided in Supplementary Tab.16-17.
For the live experiment, the neurons were incubated for 8 min with SiR-actin (0.8 μM, Cytoskeleton, cat. CY-SC001) diluted in HEPES buffered artificial cerebrospinal fluid (aCSF, in mM: NaCl 98, KCl 5, HEPES 10, glucose 10, CaCl2 0.6, MgCl2 5). For live-cell STED microscopy, coverslips were mounted on a QR chamber (Wagner Instruments, cat. 61-1944) and imaged in HEPES buffered aCSF 5 mM Mg2+ / 0.6 mM Ca2+ using a gravity perfusion system.

Kidney epithelial cell culture
For indirect immunostaining, cells were fixed in 8% paraformaldehyde (PFA) in phosphate-buffered saline (PBS) and permeabilized with 0.5% Triton X-100/PBS for Nuclear Pore Complex (NPC) proteins (Mab414, 1:200; abcam, code: ab24609) and Golgi (Giantin, 1:200; abcam, code: ab80864) staining. Methanol was used as a fixative for Tubulin staining (1:500; abcam, code: ab18251).
After blocking with 2% BSA/0.1% Tween20/PBS, cells were incubated with the primary antibody for 1 hour at the specified dilutions. Detection of primary antibodies was achieved using secondary STAR RED goat anti-mouse IgG (1:200, abberior GmbH, code: STRED-1001-500UG) and STAR ORANGE goat anti-rabbit IgG (1:200, abberior GmbH, code: STORANGE-1002-500UG) for double staining of the NPC and Golgi. Tubulin was labeled with STAR RED goat anti-rabbit IgG (1:200, abberior GmbH, code: STRED-1002-500UG). Secondary antibodies were also incubated for 1 hour.

Validation

Neuronal cell culture
Durand, A. et al. A Machine Learning Approach for Online Automated Optimization of Super-Resolution Optical Microscopy. Nature Communications 9, 5247 (Dec. 2018).
Lavoie-Cardinal, F. et al. Neuronal Activity Remodels the F-actin Based Submembrane Lattice in Dendrites but Not Axons of Hippocampal Neurons. Scientific Reports 10, 11960 (July 2020).
Wiesner, T. et al. Activity-Dependent Remodelling of Synaptic Protein Organization Revealed by High Throughput Analysis of STED Nanoscopy Images. Frontiers in Neural Circuits 14. (2020) (2020).

Kidney epithelial cell culture
Wurm, C. A., Neumann, D., Schmidt, R., Egner, A. & Jakobs, S. Sample preparation for STED microscopy. Live Cell Imaging: Methods and Protocols, 185–199 (2010).

Eukaryotic cell lines

Policy information about cell lines and Sex and Gender in Research

Cell line source(s)	DSMZ-German Collection of Microorganisms and Cell Cultures
Authentication	No authentication step was performed on our end.
Mycoplasma contamination	Negative in DAPI, microbiological culture, RNA hybridization, PCR assays (performed at DSMZ)
Commonly misidentified lines (See ICLAC register)	N/A

Animals and other research organisms

Policy information about studies involving animals; ARRIVE guidelines recommended for reporting animal research, and Sex and Gender in Research

Laboratory animals	Neuronal cultures were prepared from neonatal Sprague Dawley rats. The rats were aged P0-P1.
Wild animals	This study did not involve wild animals
Reporting on sex	The study did not involve sex-based analyses.
Field-collected samples	The study did not require field-collected samples.
Ethics oversight	The procedures for neuronal cell cultures from P0-P1 rats were approved by the animal care committee of Université Laval.

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Plants

Seed stocks	N/A
Novel plant genotypes	N/A
Authentication	N/A