ARTICLES
https://doi.org/10.1038/s41592-020-0853-5

nature methods

DeepSTORM3D: dense 3D localization microscopy and PSF design by deep learning

Elias Nehme[1,2], Daniel Freedman[3], Racheli Gordon[2], Boris Ferdman[2,4], Lucien E. Weiss[2], Onit Alalouf[2], Tal Naor[2], Reut Orange[2,4], Tomer Michaeli[1] and Yoav Shechtman[2,4]✉

An outstanding challenge in single-molecule localization microscopy is the accurate and precise localization of individual point emitters in three dimensions in densely labeled samples. One established approach for three-dimensional single-molecule localization is point-spread-function (PSF) engineering, in which the PSF is engineered to vary distinctively with emitter depth using additional optical elements. However, images of dense emitters, which are desirable for improving temporal resolution, pose a challenge for algorithmic localization of engineered PSFs, due to lateral overlap of the emitter PSFs. Here we train a neural network to localize multiple emitters with densely overlapping Tetrapod PSFs over a large axial range. We then use the network to design the optimal PSF for the multi-emitter case. We demonstrate our approach experimentally with super-resolution reconstructions of mitochondria and volumetric imaging of fluorescently labeled telomeres in cells. Our approach, DeepSTORM3D, enables the study of biological processes in whole cells at timescales that are rarely explored in localization microscopy.

Determining the nanoscale positions of point emitters forms the basis of localization microscopy techniques such as single-particle tracking[1,2], (fluorescence) photoactivated localization microscopy (f)PALM[3,4], stochastic optical reconstruction microscopy (STORM)[5] and related single-molecule localization microscopy (SMLM) methods. These techniques have revolutionized biological imaging, revealing cellular processes and structures at the nanoscale[6]. Notably, most samples of interest extend in three dimensions, necessitating three-dimensional (3D) localization microscopy[7].

In a standard microscope, the precise z position of an emitter is difficult to ascertain because the change of the PSF near the focus is approximately symmetric. Furthermore, outside of this focal range ($\approx\pm 350$ nm for a high numerical aperture imaging system), the rapid defocusing of the PSF reduces the signal-to-noise ratio (SNR), causing localization precision to quickly degrade. One method to extend the useful z range and explicitly encode the z position is PSF engineering[8–10]. Here an additional optical element, for example a phase mask, is placed in the emission path of the microscope, modifying the image formed on the detector[11] (Fig. 1a); the axial position can then be recovered via image processing using a theoretical or experimentally calibrated PSF model[10–16].

In practically all applications, it is desirable to be able to localize nearby emitters simultaneously. For example, in super-resolution SMLM experiments, the number of emitters localized per frame determines temporal resolution. In tracking applications, PSF overlap from multiple emitters often precludes localization, potentially biasing results in emitter-dense regions. The problem is that localizing overlapping emitters poses a great algorithmic challenge even in two-dimensional (2D) localization and much more so in 3D. Specifically, encoding the axial position of an emitter over large axial ranges (>3 μm) requires the use of laterally large PSFs, for example the Tetrapod[10,17] (Fig. 1b), increasing the possibility of overlap. Consequently, while a variety of methods have been developed to cope with overlapping emitters for the in-focus, standard PSF[18–20], a recent comparison of state-of-the-art software revealed that performance in high-density 3D localization situations is far from satisfactory, even for top-performing algorithms[21].

Deep learning has proven to be adept at analyzing microscopic data[22–30], especially for single-molecule localization, handling dense fields of emitters over small axial ranges (<1.5 μm)[20,31–38] or sparse emitters spread over larger ranges[39]. Moreover, an emerging application is to jointly design the optical system alongside the data processing algorithm, enabling end-to-end optimization of both components[37,40–47]. Here we present DeepSTORM3D, consisting of two fundamental contributions to high-density 3D localization microscopy over large axial ranges. First, we employ a convolutional neural network (CNN) for analyzing dense fields of overlapping emitters with engineered PSFs, demonstrated with the large-axial-range Tetrapod PSF[10,17]. Second, we design an optimal PSF for 3D localization of dense emitters over a large axial range of 4 μm. By incorporating a physical simulation layer in the CNN with an adjustable phase modulation, we jointly learn the optimal PSF (encoding) and associated localization algorithm (decoding). This approach is highly flexible and easily adapted for any 3D SMLM dataset parameters (emitter density, SNRs and z range). We quantify the performance of the method by simulation and demonstrate the applicability to 3D biological samples (mitochondria and telomeres).

Results

Dense 3D localization with DeepSTORM3D. To solve the high-density localization problem in 3D, we trained a CNN that receives a 2D image of overlapping Tetrapod PSFs spanning an axial range of 4 μm and outputs a 3D grid with a voxel size of $27.5\times27.5\times33$ nm^3 (Fig. 1c). For architecture details and learning hyper-parameters see Supplementary Notes 2.1 and 4. To compile a list of localizations, we apply simple thresholding, local maximum finding and local averaging on the output 3D grid (Supplementary Note 5).

[1]Department of Electrical Engineering, Technion, Haifa, Israel. [2]Department of Biomedical Engineering, Lorry I. Lokey Center for Life Sciences and Engineering, Technion, Haifa, Israel. [3]Google Research, Haifa, Israel. [4]Russel Berrie Nanotechnology Intitute, Technion, Haifa, Israel.
✉e-mail: yoavsh@bm.technion.ac.il

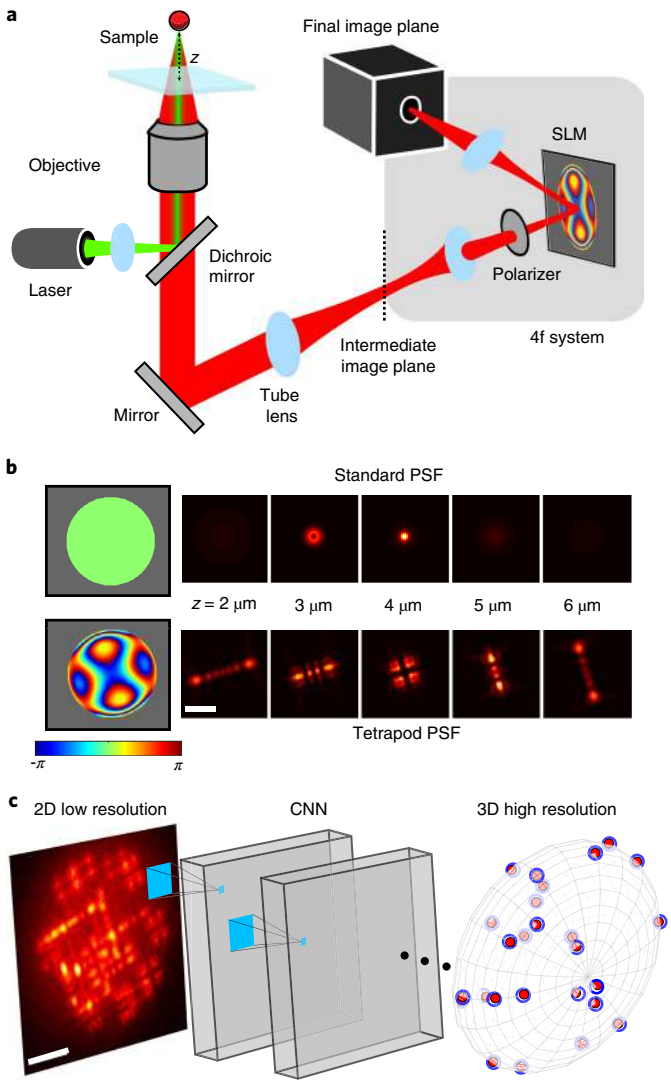


Fig. 1 | Optical setup and approach overview. a, The light emitted from a fluorescent microscopic particle is collected by the objective and focused through the tube lens into an image at the intermediate image plane. This plane is extended using a 4f system with a phase mask placed at the Fourier plane in between the two 4f lenses. **b**, The implemented phase mask (using either a liquid crystal spatial light modulator (LC-SLM) or fabricated fused silica) dictates the shape of the PSF as a function of the emitter's axial position. **c**, After training, our CNN receives a 2D low-resolution image of overlapping PSFs and outputs a 3D high-resolution volume, which is translated to a list of 3D localizations. Blue empty spheres denote simulated ground truth positions along the surface of an ellipsoid. Red spheres denote CNN detections. The Tetrapod PSF is depicted here; however, the approach is applicable to any PSF, including those optimized by the net itself (Fig. 4). Scale bars, 3 μm.

We compare our method to a fit-and-subtract-based matching pursuit (MP) approach[48] (see Supplementary Note 7) as we are unaware of any other methods capable of localizing overlapping Tetrapod PSFs. To quantitatively compare our method with MP solely in terms of density, we simulated emitters with high signal-to-noise ratio (30,000 signal counts, 150 background counts per pixel) at ten different densities ranging from 1 to 75 emitters per $13\times13\,\mu m^2$ field of view (FOV) (for the definition of density see Supplementary Note 1). The results are shown in Fig. 2. As evident in both the Jaccard index[21] (see Supplementary Note 6) and the lateral/axial root mean square error (RMSE) (Fig. 2a), the CNN achieves remarkable performance in localizing high-density Tetrapods. In the single-emitter (very low density) case, where the performance of the CNN is bounded by the discretization on the 3D grid, the RMSE of the MP localization is lower (better). This is because for a single emitter, MP is equivalent to a continuous maximum likelihood estimator (MLE) (Supplementary Note 7), which is asymptotically optimal[49], whereas the CNN's precision is bounded by pixilation of the grid (half voxel of 13.75 nm in *xy* and 16.5 nm in *z*). However, quickly beyond the single-emitter case, the CNN drastically outperforms MP at both high and low SNR (see Supplementary Note 7.3). A similar result was obtained when compared to a leading single-emitter fitting method[14] applicable also for the multiple-emitter case[21] (see Supplementary Note 8.2). Furthermore, to put our method in context with other existing approaches, we tested DeepSTORM3D on the EPFL Double Helix high-density challenge[21] obtaining favorable results (see Supplementary Note 8.1).

Next, we validated our method for super-resolution imaging of fluorescently labeled mitochondria in COS7 cells (Fig. 3 and Supplementary Videos 1–3). We acquired 20,000 diffraction-limited frames of a $50\times30\,\mu m^2$ FOV and localized them using the CNN in ≈3 h 20 m, resulting in ≈360,000 localizations. The Tetrapod PSF was implemented using a fabricated fused-silica phase mask (see Supplementary Note 9.1) and the CNN was trained solely on simulations matching the experimental conditions (see Supplementary Note 4.1). The estimated resolution was ≈40 nm in *xy* and ≈50 nm in *z* (see Supplementary Note 9.3). To visually evaluate localization performance in a single frame (Fig. 3a), we regenerated the corresponding 2D low-resolution image and overlaid the recovered image on top of the experimental frame (Fig. 3a and Supplementary Video 1). As seen in the overlay image, the emitter PSFs (3D positions) are faithfully recovered by the CNN. Emitters with an extremely low number of signal photons were ignored. For further acceleration in acquisition time see Supplementary Note 9.2.

Optimal PSF design for dense 3D imaging. The Tetrapod is a special PSF that has been optimized for the single-emitter case by Fisher information maximization[10,17]. However, when considering the multiple-emitter case, an intriguing question arises: what is the optimal PSF for high-density 3D localization over a large axial range? To answer this question we need to rethink the design metric; extending the Fisher information criterion[10] to account for emitter density is not trivial and while it is intuitive that a smaller-footprint PSF would be preferable for dense emitters, it is not clear how to mathematically balance this demand with the requirement for high localization precision per emitter.

Our PSF design logic is based on the following: as we have already established that a CNN yields superior reconstruction for high-density 3D localization, we are interested in a PSF (encoder) that would be optimally localized by a CNN (decoder). Therefore, in contrast to a sequential paradigm where the PSF and the localization algorithm are optimized separately, we adopt a co-design approach (Fig. 4a). To jointly optimize the PSF and the localization CNN, we introduced a differentiable physical simulation layer, which is parametrized by a phase mask that dictates the microscope's PSF. This layer encodes 3D point sources to their respective low-resolution 2D image (see Supplementary Note 3). This image is then fed to the localization CNN, which decodes it and recovers the underlying 3D source positions. During training, the net is presented with simulated point sources at random locations (projected on a fine grid) and using the difference between the CNN recovery and the simulated 3D positions quantified by our loss function (see Supplementary Note 4.2), we optimize both the phase mask and the localization CNN parameters in an end-to-end fashion using the backpropagation algorithm[50] (see Supplementary Note 3.3 and Supplementary Video 4). The learned PSF (Fig. 4b) has a small

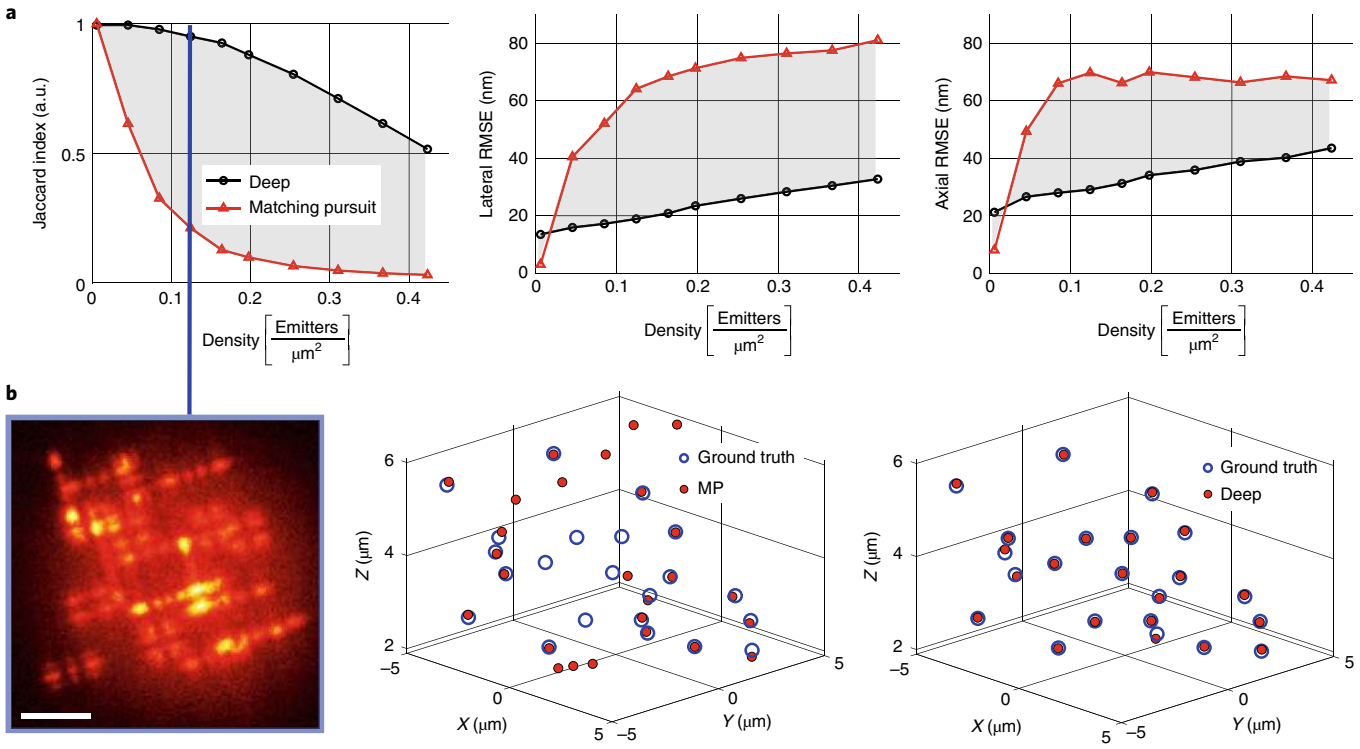


Fig. 2 | Comparison to MP. a, Performance comparison of a trained CNN (black) and the MP approach (red) in both detectability (Jaccard index) and in precision (lateral\axial RMSE). Matching of points was computed with a threshold distance of 150 nm using the Hungarian algorithm. Each data point is an average of $n = 100$ simulated images. Average s.d. in Jaccard index was ≈6% for both methods and average s.d. in precision was ≈6 nm for the CNN and ≈15 nm for MP. **b**, Example of a simulated frame of density 0.124 $\left[\frac{\text{emitters}}{\mu m^2}\right]$ alongside 3D comparisons of the recovered positions by MP (middle) and by the CNN (right). Scale bar, 2 μm.

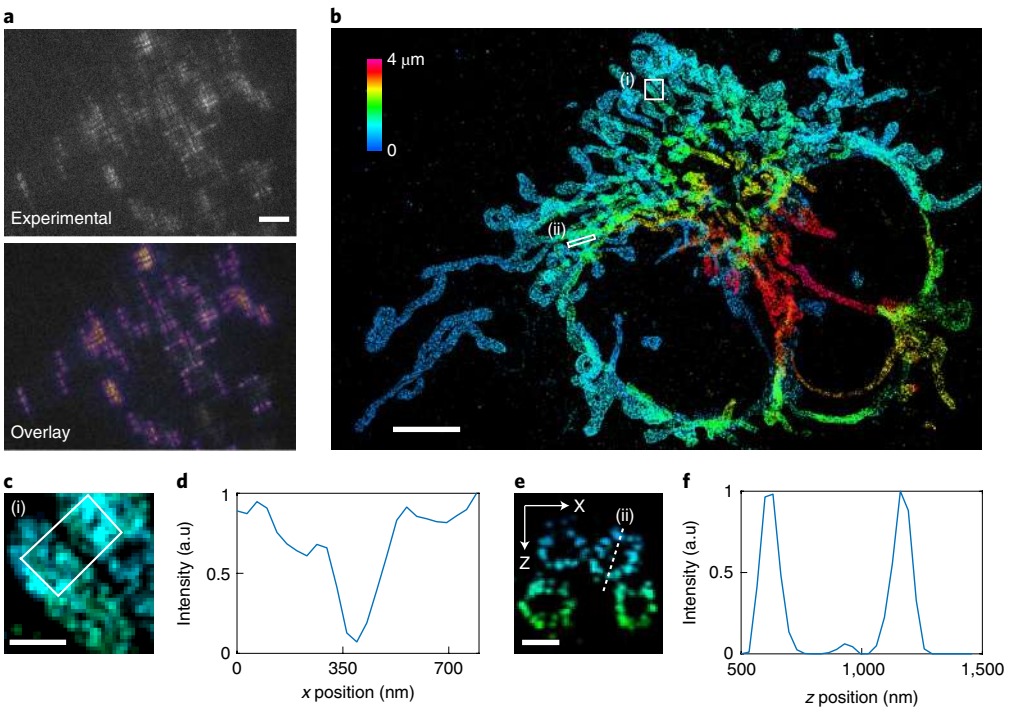


Fig. 3 | Super-resolution 3D imaging over a 4 μm z range. a, Representative experimental frame (top) and rendered frame from the 3D recovered positions by the CNN overlaid on top (bottom). Scale bar, 5 μm. **b**, Super-resolved image of mitochondria spanning a ≈4 μm *z* range rendered as a 2D histogram, where *z* is encoded by color. Scale bar, 5 μm. **c**, Zoom in on white rectangle (i) in **b**. Scale bar, 0.5 μm. **d**, Relative intensity averaged along the shorter side of the white rectangle in **c**. **e**, *XZ* cross-section of white rectangle (ii) in **b**. Scale bar, 0.5 μm. **f**, Relative intensity along the dashed white line in **e**. The experiment was repeated independently for $n = 3$ cells, twice analyzing 20,000 frames and once analyzing 10,000 frames all leading to similar performance.

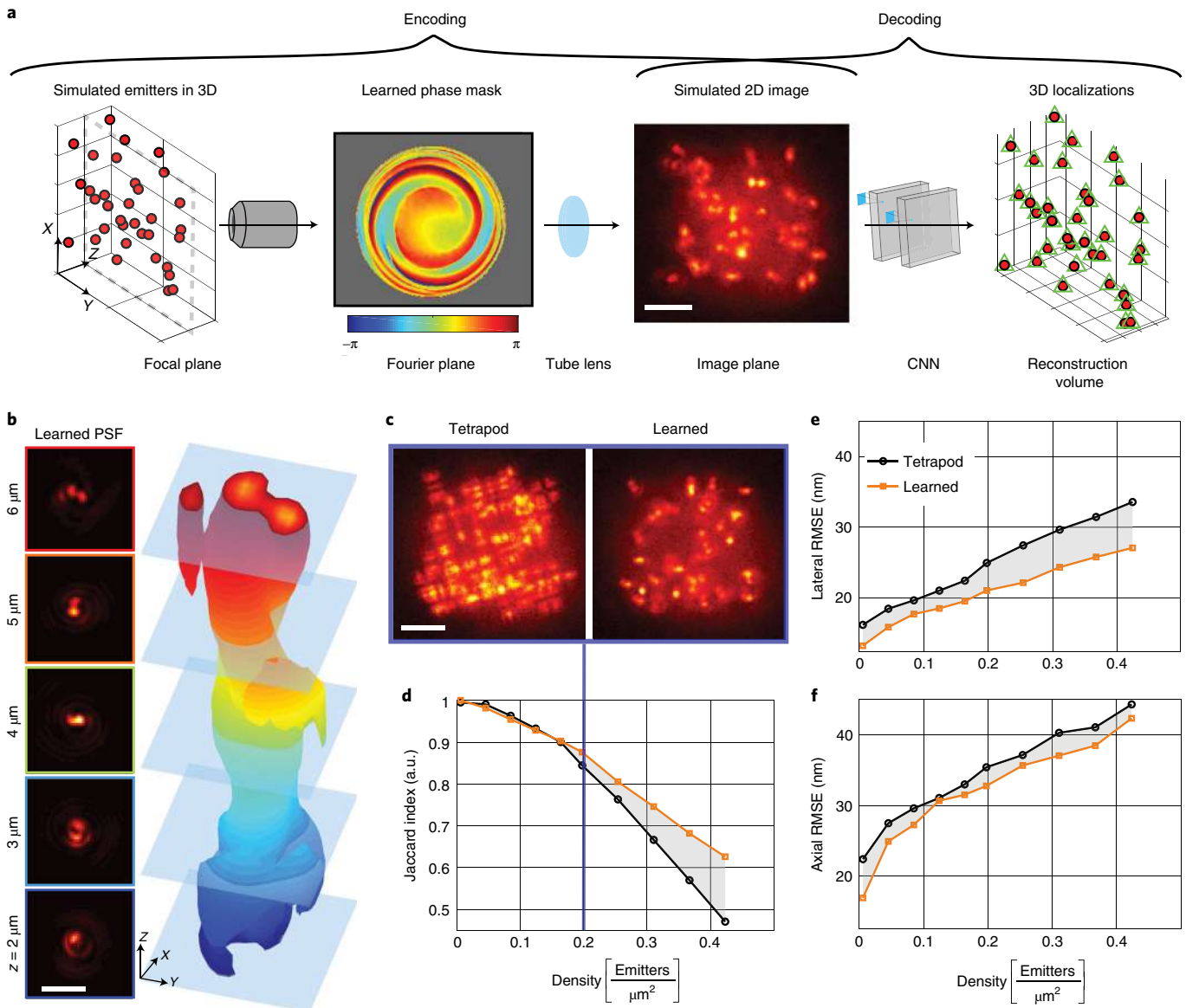


Fig. 4 | PSF learning for high-density 3D imaging. a, Simulated 3D emitter positions are fed to the image formation model to simulate their low-resolution camera image (encoding). Next, this image is fed to a CNN that tries to recover the simulated emitter positions (decoding). The difference between the simulated positions and the positions recovered by the CNN is used to jointly optimize the phase mask at the Fourier plane and the recovery CNN parameters. **b**, Simulation of the learned PSF as function of the emitter axial position (left). 3D isosurface rendering of the learned PSF (right). **c**, Example frame of density 0.197 $\left[\frac{\text{emitters}}{\mu m^2}\right]$ with the same simulated emitter positions, using the Tetrapod (left) and the learned PSF (right). **d**, Jaccard index comparison between two CNNs with the same architecture, one trained to recover 3D positions from 2D images of Tetrapod PSF (black) and the second trained to recover 3D positions from 2D images of the learned PSF (orange). Each data point is an average of $n = 100$ simulated images. Average s.d. was ≈6% for both PSFs. **e**,**f**, Lateral and axial RMSE comparison between the same two CNNs from **d**. Average s.d. was ≈6 nm for both PSFs. Scale bars, 3 µm.

lateral footprint, which is critical for minimizing overlap at high densities. The learned phase mask twists in a spiral trajectory causing the PSF to rapidly rotate throughout the axial range, a trait that was previously shown to be valuable for encoding depth[8].

To quantify the improvement introduced by our new PSF, we first compared it to the Tetrapod PSF in simulations. Specifically, we trained a similar reconstruction net for both the Tetrapod and the learned PSF using a matching training set composed of simulated continuous 3D positions along with their corresponding 2D low-resolution images. The learned PSF performs similarly to the Tetrapod PSF for low emitter densities (Fig. 4d–f). However, as the density goes up (higher than ≈0.2 $\left[\frac{\text{emitters}}{\mu m^2}\right]$) the learned PSF outperforms the Tetrapod PSF in both localization precision (Fig. 4e,f) and in emitter detectability (Jaccard index) (Fig. 4d). This result is not surprising, as the learned PSF has a smaller spatial footprint and hence it is less likely to overlap than the Tetrapod (Fig. 4c). For further analysis of the learned PSF see Supplementary Note 10.

Volumetric telomere imaging and tracking. Next, we demonstrate the superiority of the new PSF experimentally by imaging fluorescently labeled telomeres (DsRed-hTRF1) in fixed U2OS cells. The cell contains tens of telomeres squeezed in the volume of a nucleus with ≈20 µm diameter (Fig. 5a,b). From a single snapshot focused inside the nucleus, the CNN outputs a list of 3D positions of telomeres spanning an axial range of ≈3 µm. Using the Tetrapod PSF snapshot, the Tetrapod-trained CNN was able to recover 49 out of 62 telomeres with a single false positive, yielding a Jaccard index

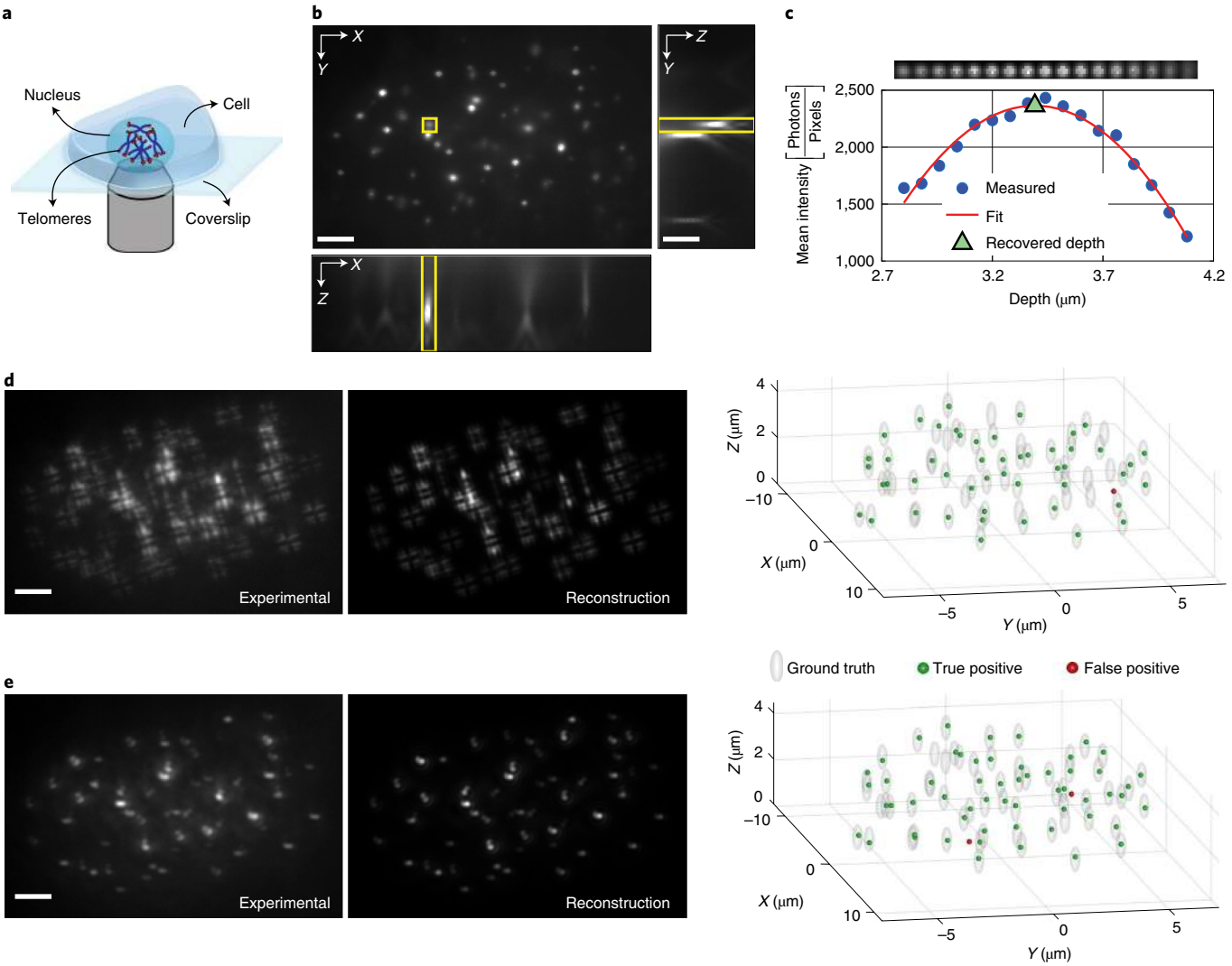


Fig. 5 | Three-dimensional imaging of telomeres in a single snapshot. a, Schematic of imaging fixed U2OS cells with fluorescently labeled telomeres inside their nucleus. **b**, Focus slice with the standard PSF inside a U2OS cell nucleus, obtained via a *z* scan. The yellow rectangles mark the same emitter in all three orthogonal planes. **c**, Example fit of the mean intensity in sequential axial slices used to estimate approximate emitter axial position. **d**, Experimental snapshot with the Tetrapod PSF (left), rendered image from the 3D recovered positions by the Tetrapod CNN (middle) and a 3D comparison of the recovered positions and the approximate experimental ground truth (right). **e**, Experimental snapshot with the learned PSF (left), rendered image from the 3D recovered positions by the learned PSF CNN (middle) and a 3D comparison of the recovered positions and the approximate experimental ground truth (right). Scale bars, 3 μm. The experiment was repeated independently for $n = 10$ U2OS cells all showing similar characteristics and performance.

of 0.77 (Fig. 5d). In comparison, using the learned PSF snapshot, the corresponding CNN was able to recover 57 out of the 62 telomeres with only two false positives, yielding a Jaccard index of 0.89 (Fig. 5e). The recovered positions were compared to approximated ground-truth 3D positions (Fig. 5c), obtained by axial scanning and 3D fitting (see Supplementary Note 12 and Supplementary Video 5). The precision of both PSFs was calibrated experimentally using a *z* scan of a fluorescent microsphere (see Supplementary Note 10.4 and Supplementary Video 6).

To qualitatively compare the recovered list of localizations to the acquired snapshot, we fed this list to the physical simulation layer and generated the matching 2D low-resolution image (Fig. 5d,e). As verified by the regenerated images, the 3D positions of the telomeres are faithfully recovered by the CNNs. Moreover, the misses in both snapshots were either due to local aberrations and/or an extremely low number of signal photons (see Supplementary Note 13.1 for more experimental results).

Finally, a great advantage facilitated by our scan-free learned PSF is increased volumetric temporal resolution. To demonstrate the full capability of our method for dense multiparticle localization, we simultaneously tracked 48 telomeres, diffusing within the volume of a live mouse embryonic fibroblast (MEF) cell, at 10 Hz over 50 s (Fig. 6). Such a measurement can provide information on 3D nuclear rotation (Fig. 6a and Supplementary Video 7) and heterogeneity in motion type (Fig. 6b–d), at timescales that are typically unexplored by volumetric localization microscopy[51].

Discussion

In this work we demonstrated 3D localization of dense emitters over a large axial range, both numerically and experimentally. The described network architecture exhibits excellent flexibility in dealing with various experimental challenges, for example low signal-to-noise ratios and optical aberrations. This versatility is facilitated in three ways: (1) the net was trained solely on simulated

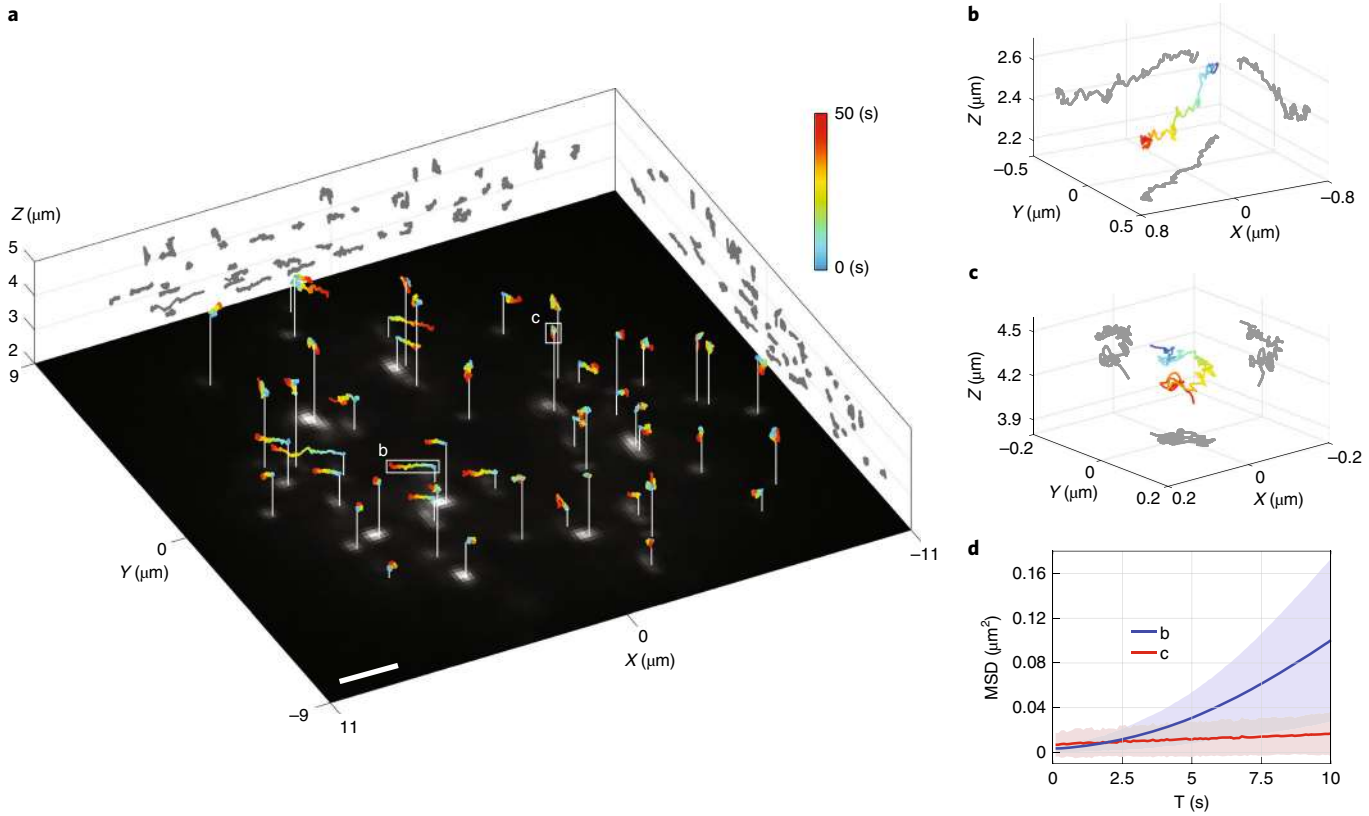


Fig. 6 | Volumetric tracking of telomeres in live MEF cells. a, 3D trajectories of tracked telomeres on top of the first experimental snapshot. White sticks mark the starting point and color encodes time. *XZ* and *YZ* projections are plotted in gray. Scale bar, 2 μm. **b**, Example trajectory of a telomere (white box in **a**) with a large mean displacement. **c**, Example trajectory of a telomere (white box in **a**) with a small mean displacement. **d**, Mean squared displacement (MSD) of the trajectories in **b** and **c**. Shaded area marks one s.d. The experiment was repeated independently for $n = 10$ MEF cells all showing similar characteristics and performance.

data, thus producing sufficiently large datasets for optimization; (2) the phase mask that governs the PSF was optimized with respect to the implementation in the imaging system, that is the pixels of the spatial light modular, rather than over a smaller subspace, for example Zernike polynomials[10]; and (3) the CNN localization algorithm was designed in coordination with the development of the PSF, thus the system was optimized for the desired output[37] rather than a proxy.

Attaining a sufficiently large training dataset has thus far been a major limitation for most applications of CNNs. With this limitation in mind, the application of CNNs to single-molecule localization would seemingly be an ideal one, as each emitter's behavior should be approximately the same. This uniformity is broken, however, by spatially varying background, sample density and variable emitter size in biological samples (Supplementary Note 4.1), all of which diversify datasets and necessitate relevant training data. By implementing an accurate simulator (Supplementary Note 3), we have shown that it is possible to build a robust network entirely in silico, generating arbitrarily large, realistic datasets with a known ground truth to optimize nets. This aligns with our previous work in 2D SMLM[20].

For super-resolution reconstructions using the Tetrapod PSF, the simulator was particularly important due to the highly variable SNR of emitters in the sample. Here, our net was able to selectively localize emitters even in very dense regions by focusing on those with a high SNR (Fig. 3). To optimize a PSF while simultaneously training the net, the simulator was also essential, as it would be prohibitively time consuming to experimentally vary the PSF, while recording and analyzing images to train the net.

A notable aspect of our optimization approach is that the optimized PSF is found by continuously varying the pixels of an initialized mask, while evaluating the output of the localization net, thus the final result represents a local minimum (Fig. 4). By changing the initialization conditions, we have recognized several patterns that indicate how the optimal PSF varies with the experimental conditions, namely, density, axial range and SNR (see Supplementary Note 2.2). Some of the recurrent features are intuitive: for example, in dense fields of emitters with limited SNR, the optimized PSFs have a small footprint over the designed axial range, enabling high density and compacting precious signal photons into as few pixels as possible. What distinguishes the net PSFs over predetermined designs is the utilization of multiple types of depth encoding, namely, simultaneously employing astigmatism, rotation and side lobe movement (Fig. 4), all of which have been conceived of and implemented previously, but never simultaneously.

This work, therefore, triggers many possible questions and research directions regarding its capabilities and limitations. For example, how globally optimal is the resulting PSF? Similarly, how sensitive is the resulting PSF and its performance to different loss functions, CNN architectures, initializations (for example with an existing phase mask) and the sampled training set of locations? Currently, it is unclear how each of these components affects the learning process, although we began to partially answer them in simulations (see Supplementary Notes 2.2 and 10). Finally, the co-design approach employed here paves the way to a wide variety of interesting applications in microscopy, where imaging systems have traditionally been designed separately from the processing algorithm.

Online content

Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41592-020-0853-5.

Received: 10 September 2019; Accepted: 6 May 2020;
Published online: 15 June 2020

References

1. Katayama, Y. et al. Real-time nanomicroscopy via three-dimensional single-particle tracking. *Chem. Phys. Chem.* **10**, 2458–2464 (2009).
2. Manzo, C. & Garcia-Parajo, M. F. A review of progress in single particle tracking: from methods to biophysical insights. *Rep. Prog. Phys.* **78**, 124601 (2015).
3. Betzig, E. et al. Imaging intracellular fluorescent proteins at nanometer resolution. *Science* **313**, 1642–1645 (2006).
4. Hess, S. T., Girirajan, T. P. & Mason, M. D. Ultra-high resolution imaging by fluorescence photoactivation localization microscopy. *Biophysical J.* **91**, 4258–4272 (2006).
5. Rust, M. J., Bates, M. & Zhuang, X. Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (storm). *Nat. Methods* **3**, 793–796 (2006).
6. Sahl, S. J. & Moerner, W. Super-resolution fluorescence imaging with single molecules. *Curr. Opin. Struct. Biol.* **23**, 778–787 (2013).
7. von Diezmann, A., Shechtman, Y. & Moerner, W. Three-dimensional localization of single molecules for super-resolution imaging and single-particle tracking. *Chem. Rev.* **117**, 7244–7275 (2017).
8. Pavani, S. R. P. et al. Three-dimensional, single-molecule fluorescence imaging beyond the diffraction limit by using a Double-Helix point spread function. *Proc. Natl Acad. Sci. USA* **106**, 2995–2999 (2009).
9. Huang, B., Wang, W., Bates, M. & Zhuang, X. Three-dimensional super-resolution imaging by stochastic optical reconstruction microscopy. *Science* **319**, 810–813 (2008).
10. Shechtman, Y., Sahl, S. J., Backer, A. S. & Moerner, W. Optimal point spread function design for 3D imaging. *Phys. Rev. Lett.* **113**, 133902 (2014).
11. Backer, A. S. & Moerner, W. Extending single-molecule microscopy using optical Fourier processing. *J. Phys. Chem. B* **118**, 8313–8329 (2014).
12. Liu, S., Kromann, E. B., Krueger, W. D., Bewersdorf, J. & Lidke, K. A. Three-dimensional single-molecule localization using a phase retrieved pupil function. *Opt. express* **21**, 29462–29487 (2013).
13. Babcock, H. P. & Zhuang, X. Analyzing single molecule localization microscopy data using cubic splines. *Sci. Rep.* **7**, 552 (2017).
14. Li, Y. et al. Real-time 3D single-molecule localization using experimental point-spread functions. *Nat. Methods* **15**, 367 (2018).
15. Aristov, A., Lelandais, B., Rensen, E. & Zimmer, C. Zola-3D allows flexible 3D localization microscopy over an adjustable axial range. *Nat. Commun.* **9**, 2409 (2018).
16. Ferdman, B. et al. VIPR: vectorial implementation of phase retrieval for fast and accurate microscopic pixel-wise pupil estimation. *Opt. Express* **28**, 10179–10198 (2020).
17. Shechtman, Y., Weiss, L. E., Backer, A. S., Sahl, S. J. & Moerner, W. Precise three-dimensional scan-free multiple-particle tracking over large axial ranges with tetrapod point spread functions. *Nano Lett.* **15**, 4194–4199 (2015).
18. Min, J. et al. Falcon: fast and unbiased reconstruction of high-density super-resolution microscopy data. *Sci. Rep.* **4**, 4577 (2014).
19. Boyd, N., Schiebinger, G. & Recht, B. The alternating descent conditional gradient method for sparse inverse problems. *SIAM J. Optim.* **27**, 616–639 (2017).
20. Nehme, E., Weiss, L. E., Michaeli, T. & Shechtman, Y. Deep-storm: super-resolution single-molecule microscopy by deep learning. *Optica* **5**, 458–464 (2018).
21. Sage, D. et al. Super-resolution fight club: assessment of 2D and 3D single-molecule localization microscopy software. *Nat. Methods* **16**, 387 (2019).
22. Rivenson, Y., Zhang, Y., Günaydın, H., Teng, D. & Ozcan, A. Phase recovery and holographic image reconstruction using deep learning in neural networks. *Light-Sci. Appl.* **7**, 17141 (2018).
23. Nguyen, T., Xue, Y., Li, Y., Tian, L. & Nehmetallah, G. Deep learning approach for Fourier ptychography microscopy. *Opt. Express* **26**, 26470–26484 (2018).
24. Weigert, M. et al. Content-aware image restoration: pushing the limits of fluorescence microscopy. *Nat. Methods* **15**, 1090 (2018).
25. Ounkomol, C., Seshamani, S., Maleckar, M. M., Collman, F. & Johnson, G. R. Label-free prediction of three-dimensional fluorescence images from transmitted-light microscopy. *Nat. Methods* **15**, 917–920 (2018).
26. Krull, A., Buchholz, T.-O. & Jug, F. Noise2void-learning denoising from single noisy images. in *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (eds. Davis, L., Torr, P. & Zhu, S. C.) 2129–2137 (2019).
27. Falk, T. et al. U-net: deep learning for cell counting, detection, and morphometry. *Nat. Methods* **16**, 67–70 (2019).
28. Rivenson, Y. et al. Phasestain: the digital staining of label-free quantitative phase microscopy images using deep learning. *Light-Sci. Appl.* **8**, 23 (2019).
29. Liu, T. et al. Deep learning-based super-resolution in coherent imaging systems. *Sci. Rep.* **9**, 3926 (2019).
30. Smith, J. T. et al. Fast fit-free analysis of complex fluorescence lifetime imaging via deep learning. *Proc. Natl Acad. Sci. USA* **116**, 24019–24030 (2019).
31. Boyd, N., Jonas, E., Babcock, H. P. & Recht, B. DeepLoco: fast 3D localization microscopy using neural networks. Preprint at *bioRxiv* https://doi.org/10.1101/267096 (2018).
32. Ouyang, W., Aristov, A., Lelek, M., Hao, X. & Zimmer, C. Deep learning massively accelerates super-resolution localization microscopy. *Nat. Biotechnol.* **36**, 460–468 (2018).
33. Diederic, B., Then, P., Jügler, A., Förster, R. & Heintzmann, R. cellSTORM: cost-effective super-resolution on a cellphone using dSTORM. *PloS ONE* **14**, e0209827 (2019).
34. Newby, J. M., Schaefer, A. M., Lee, P. T., Forest, M. G. & Lai, S. K. Convolutional neural networks automate detection for tracking of submicron-scale particles in 2D and 3D. *Proc. Natl Acad. Sci. USA* **115**, 9026–9031 (2018).
35. Zelger, P. et al. Three-dimensional localization microscopy using deep learning. *Opt. Express* **26**, 33166–33179 (2018).
36. Liu, K. et al. Fast 3D cell tracking with wide-field fluorescence microscopy through deep learning. Preprint at https://arXiv.org/abs/1805.05139 (2018).
37. Hershko, E., Weiss, L. E., Michaeli, T. & Shechtman, Y. Multicolor localization microscopy and point-spread-function engineering by deep learning. *Opt. Express* **27**, 6158–6183 (2019).
38. Speiser, A., Turaga, S. C. & Macke, J. H. Teaching deep neural networks to localize sources in super-resolution microscopy by combining simulation-based learning and unsupervised learning. Preprint at https://arXiv.org/abs/1907.00770 (2019).
39. Zhang, P. et al. Analyzing complex single-molecule emission patterns with deep learning. *Nat. methods* **15**, 913 (2018).
40. Chakrabarti, A. Learning sensor multiplexing design through back-propagation. in *Advances in Neural Information Processing Systems* (eds. Lee, D. D. et al.) 3081–3089 (Curran Associates, 2016).
41. Horstmeyer, R., Chen, R. Y., Kappes, B. & Judkewitz, B. Convolutional neural networks that teach microscopes how to image. Preprint at https://arXiv.org/abs/1709.07223 (2017).
42. Turpin, A., Vishniakou, I. & D Seelig, J. Light-scattering control in transmission and reflection with neural networks. *Opt. Express* **26**, 30911–30929 (2018).
43. Haim, H., Elmalem, S., Giryes, R., Bronstein, A. M. & Marom, E. Depth estimation from a single image using deep learned phase coded mask. *IEEE Trans. Comput. Imaging* **4**, 298–310 (2018).
44. He, L., Wang, G. & Hu, Z. Learning depth from single images with deep neural network embedding focal length. *IEEE Trans. Image Process.* **27**, 4676–4689 (2018).
45. Sitzmann, V. et al. End-to-end optimization of optics and image processing for achromatic extended depth of field and super-resolution imaging. *ACM Trans. Graph.* **37**, 114 (2018).
46. Chang, J. & Wetzstein, G. Deep optics for monocular depth estimation and 3D object detection. in *Proc. IEEE International Conference on Computer Vision* (eds. Lee, K. M. et al.) 10193–10202 (2019).
47. Wu, Y., Boominathan, V., Chen, H., Sankaranarayanan, A. & Veeraraghavan, A. Phasecam3D: learning phase masks for passive single view depth estimation. in *IEEE International Conference on Computational Photography* (ed. Nedevschi, S.) 1–12 (2019).
48. Shechtman, Y., Weiss, L. E., Backer, A. S., Lee, M. Y. & Moerner, W. Multicolour localization microscopy by point-spread-function engineering. *Nat. Photonics* **10**, 590 (2016).
49. Bickel, P. J. & Doksum, K. A. *Mathematical Statistics: Basic Ideas and Selected Topics, Volumes I-II Package* (Chapman and Hall/CRC, 2015).
50. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
51. Bronshtein, I. et al. Loss of lamin A function increases chromatin dynamics in the nuclear interior. *Nat. Commun.* **6**, 8044 (2015).

Methods

Sample preparation. COS7 cells were grown for 24 h on cleaned 22 × 22 mm, 170-µm thick coverslips in a six-well plate in DMEM with 1 g l^{-1} D-glucose (low glucose), supplemented with fetal bovine serum, penicillin–streptomycin and glutamine at 37 °C and 5% CO_2. Cells were fixed with 4% paraformaldehyde and 0.2% glutaraldehyde in PBS (pH 6.2) for 45 min, washed and incubated in 0.3 M glycine/PBS solution for 10 min. The coverslips were transferred into a clean six-well plate and incubated in a blocking solution for 2 h (10% goat serum, 3% BSA, 2.2% glycine and 0.1% Triton-X in PBS, filtered with 0.45-µm PVDF filter unit, Millex). The cells were then immunostained overnight with anti TOMM20-AF647 (Abcam, ab209606) 1:230 diluted in blocking buffer and washed five times with PBS. Cover glasses (22 × 22 mm, 170 µm thick) were cleaned in an ultrasonic bath with 5% Decon90 at 60 °C for 30 min, then washed with water, incubated in ethanol absolute for 30 min and sterilized with 70% filtered ethanol for 30 min.

U2OS cells were grown on cleaned 0.18-mm coverslips in a 12-well plate in DMEM with 1 g l^{-1} D-glucose (low glucose), supplemented with fetal bovine serum, penicillin–streptomycin and glutamine at 37 °C and 5% CO_2. The day after cells were transfected with a plasmid encoding the fluorescently tagged telomeric repeat binding factor 1 (DsRed-hTRF1)[51] using Lipofectamine 3000 reagent. At 24 h after transfection, cells were fixed with 4% paraformaldehyde for 20 min, washed three times with PBS and attached to a slide together with mounting medium.

Lamin A double knockout (lmna$^{-/-}$) MEFs were cultured in phenol red-free DMEM/F-12 medium (Gibco, Thermo Fisher Scientific), which was supplemented with 10% fetal bovine serum and 1% penicillin–streptomycin solution (Biological Industries, Bet Ha-emek). Two days before imaging, cells were transferred to 15-mm coverglass-bottom culture plates (Nest scientific). After 24 h, cells were transfected with a plasmid encoding DsRed-hTRF1. The transfection mix was prepared by diluting 8 µg of the plasmid in 50 µl of serum-free DMEM/F-12 and separately diluting 24 µl of transfection reagent Polyjet (SignaGen Laboratories) in 26 µl of serum-free DMEM/F-12, then immediately adding the diluted Polyjet solution to the DNA mixture and incubating at room temperature for 20 min. The 100 µl DNA–Polyjet mix was then added dropwise to cell plates. Imaging experiments were conducted approximately 26 h after transfection and lasted for 1–2 h.

Optical setup. Imaging experiments were performed on the experimental system shown schematically in Fig. 1a. The 4f optical processing system was built alongside the side port of a Nikon Eclipse Ti inverted fluorescence microscope, with a ×100/1.45 NA oil-immersion objective lens (Plan Apo ×100/1.45 NA, Nikon).

STORM imaging. For super-resolution imaging, a polydimethylsiloxane chamber was attached to a glass coverslip containing fixed COS7 cells. Blinking buffer (100 mM β-mercaptoethylamine hydrochloride, 20% sodium lactate and 3% OxyFluor (Sigma, SAE0059), modified from Nahidiazar et al.[52], was then added and a glass coverslip was placed on top to prevent evaporation. Low-intensity illumination for recording diffraction-limited images was applied using a Topica laser (640 nm), on the Nikon TI imaging setup described previously and recorded with an EMCCD (iXon, Andor) in a standard imaging setup. For super-resolution blinking using the Tetrapod PSF, high-intensity (1 W at the back of the objective lens) 640 nm light was applied using a 638 nm 2,000 mW red dot laser module, whose beam shape was cleaned using a 25-µm pinhole (Thorlabs) in coordination with low-intensity (<5 mW) 405 nm light. Emission light was filtered through a 500-nm long pass dichroic and a 650-nm long pass (Chroma), projected through a 4f system containing the dielectric Tetrapod phase mask (see Supplementary Note 9.1) and imaged on a Prime95b Photometrics camera.

Super-resolution image rendering. Before rendering the super-resolved image (Fig. 3b), we first corrected for sample drift using the ThunderSTORM[53] ImageJ Fiji plugin[54]. Afterward, we rendered 3D localizations as a 2D average shifted histogram, with color encoding the z position.

Telomere imaging. For telomere imaging in fixed cells, the 4f system consisted of two $f = 15$ cm lenses (Thorlabs), a linear polarizer (Thorlabs) to filter out the light that is polarized in the unmodulated direction of the LC-SLM, a 1,920 × 1,080 pixel LC-SLM (PLUTO-VIS, Holoeye) and a mirror for beam-steering. A sCMOS camera (Prime95B, Photometrics) was used to record the data. The sample was illuminated with 561-nm fiber-coupled laser light source (iChrome MLE, Toptica). The excitation light was reflected up through the microscope objective by a multibandpass dichroic filter (TRF89902-EM-ET-405/488/561/647 nm Laser Quad Band Set, Chroma). Emission light was filtered by the same dichroic and also filtered by another 617-nm band pass filter (FF02-617/73, Semrock).

For volumetric telomere tracking in live cells, images were recorded with an EMCCD camera (Andor iXON), exposure time of 100 ms and EM-gain of 170. The sample was illuminated at ≈2 kW cm^{-2} with 561 nm light from a fiber-coupled laser (iChrome MLE, Toptica). All movies were recorded for 50 s (500 frames).

CNN architecture. In summary, our localization CNN architecture is composed of three main modules. First, a multiscale context aggregation module processes the input 2D low-resolution image and extracts features with a growing receptive field using dilated convolutions[55]. Second, an upsampling module increases the lateral resolution of the predicted volume by fourfold. Finally, the last module refines the depth and lateral position of the emitters and outputs the predicted vacancy grid. For more details regarding the architecture see Supplementary Note 2.

Statistics and reproducibility. The STORM experiment was repeated independently for $n = 3$ cells, twice analyzing 20,000 frames and once analyzing 10,000 frames, all leading to similar performance. The fixed telomere experiment was repeated independently for $n = 10$ U2OS cells all showing similar characteristics and performance. The live telomere experiment was repeated independently for $n = 10$ MEF cells all showing similar characteristics and performance.

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

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Code availability

Code is made publicly available at https://github.com/EliasNehme/DeepSTORM3D.

References

52. Nahidiazar, L., Agronskaia, A. V., Broertjes, J., van den Broek, B. & Jalink, K. Optimizing imaging conditions for demanding multi-color super resolution localization microscopy. *PLoS ONE* **11**, e0158884 (2016).
53. Ovesný, M., Křížek, P., Borkovec, J., Švindrych, Z. & Hagen, G. M. ThunderSTORM: a comprehensive ImageJ plug-in for PALM and STORM data analysis and super-resolution imaging. *Bioinformatics* **30**, 2389–2390 (2014).
54. Schindelin, J. et al. Fiji: an open-source platform for biological-image analysis. *Nat. Methods* **9**, 676 (2012).
55. Yu, F. & Koltun, V. Multi-scale context aggregation by dilated convolutions. Preprint at https://arXiv.org/abs/1511.07122v3 (2016).

Acknowledgements

We thank the Garini laboratory (Bar-Ilan University) for the U2OS cells, lmna$^{-/-}$ MEFs and the plasmid encoding for DsRed-hTRF1. We thank J. Ries for help with the application of SMAP-2018 to Tetrapod PSFs. We gratefully acknowledge the support of the NVIDIA Corporation with the donation of the Titan V GPU used for this research. We thank the staff of the Micro-Nano-Fabrication and Printing Unit at the Technion for their assistance with the phase mask fabrication. We thank Google for the research cloud units provided to accelerate this research. E.N., O.A., B.F. and R.O. are supported by H2020 European Research Council Horizon 2020 (802567); T.M. is supported by the Israel Science Foundation (grant no. 852/17) and by the Ollendorff Foundation; R.G. and O.A. are supported by the Israel Science Foundation (grant no. 450/18); Y.S. is supported by the Technion-Israel Institute of Technology Career Advancement Chairship; L.E.W. and Y.S. are supported by the Zuckerman Foundation. D.F. is supported by Google.

Author contributions

E.N., D.F., T.M. and Y.S. conceived the approach. E.N. performed the simulations and analyzed the data with contributions from all authors. E.N., R.G., B.F., L.E.W., O.A. and T.N. collected the data. R.O. fabricated the physical phase mask. T.N. prepared MEF cells. E.N., D.F., L.E.W., T.M. and Y.S. wrote the paper with contributions from all authors.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41592-020-0853-5.

Correspondence and requests for materials should be addressed to Y.S.

Peer review information Rita Strack was the primary editor on this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Reprints and permissions information is available at www.nature.com/reprints.

nature research

Corresponding author(s): Yoav Shechtman

Last updated by author(s): Jun 23, 2020

Reporting Summary

Nature Research 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 Research policies, see Authors & Referees 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	Nikon Imaging Software v 5.02.02
Data analysis	Matlab 2017b, Python 3.6. A list of exact python package versions are present on GitHub within the environment.yml file.

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/reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research 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 list of figures that have associated raw data
- A description of any restrictions on data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

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	Sample size was not predetermined based off statistical calculations. The mitochondria experiment was repeated independently for 3 cells, twice analyzing 20K frames and once analyzing 10K frames to ensure reproducibility. For telomere samples we imaged two different types of cells (U2OS and MEF) on two independent setups (different cameras and SLMs) in order to ensure we cover the range of possible biological as well as experimental variations. The variation between different experiments was low.
Data exclusions	No data was excluded.
Replication	All replication attempts were successful.
Randomization	No randomization was used for experimental data; however, simulations systematically test a number of different imaging conditions. No sample randomization was performed in this study.
Blinding	Analyzed data for this study is illustrative of potential applications and was not used to extract specific biological conclusions, thus blinding was not relevant to this study.

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
☒	☐ Animals and other organisms
☒	☐ Human research participants
☒	☐ Clinical data

Methods

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

Antibodies

Antibodies used	anti TOMM20-AF647 (dilution 1:230, rabbit monoclonal, ab209606, Lot CR3270643-1, Abeam)
Validation	According to the manufacturer anti TOMM20-AF647 antibody (rabbit monoclonal, ab209606, Abcam) is reactive against Homo sapiens (Human) and was validated in Hela cells for immunocytochemistry and immunofluorescence.

Eukaryotic cell lines

Policy information about cell lines

Cell line source(s)	COS7 cells were a gift of the Elia lab (Ben-Gurion University), U2OS cells were a kind gift of the Garini lab (Bar-Ilan University) used in ref [51], MEF la min A double knockout cells were also a kind gift of the Garini lab (Bar-I Ian University) used in ref [51].
Authentication	none of the cell lines used were authenticated.
Mycoplasma contamination	Only MEF lamin A double knockout cells were tested and found negative for mycoplasma contamination.
Commonly misidentified lines (See ICLAC register)	No commonly misidentified cell lines were used.

nature methods

SUPPLEMENTARY INFORMATION

https://doi.org/10.1038/s41592-020-0853-5

In the format provided by the authors and unedited.

DeepSTORM3D: dense 3D localization microscopy and PSF design by deep learning

Elias Nehme[1,2], Daniel Freedman[3], Racheli Gordon[2], Boris Ferdman[2,4], Lucien E. Weiss[2], Onit Alalouf[2], Tal Naor[2], Reut Orange[2,4], Tomer Michaeli[1] and Yoav Shechtman[2,4] ✉

[1]Department of Electrical Engineering, Technion, Haifa, Israel. [2]Department of Biomedical Engineering, Lorry I. Lokey Center for Life Sciences and Engineering, Technion, Haifa, Israel. [3]Google Research, Haifa, Israel. [4]Russel Berrie Nanotechnology Intitute, Technion, Haifa, Israel.
✉e-mail: yoavsh@bm.technion.ac.il

Supplementary notes

Contents

1 Density definition

Usually when discussing volumetric density it is standard to define density as the number of emitters divided by the volume. However, when these emitters are imaged onto a 2D sensor using depth-encoding PSFs (e.g. [1, 2]), this definition is misleading and does not intuitively reflect the difficulty of localizing the emitters in 3D. This is due to the fact that larger axial ranges usually require PSFs with a larger lateral extent [3], which means it is actually harder to localize the same number of emitters over a larger axial range as their measured PSFs are more likely to overlap on the 2D sensor. On the other hand, if we define density in 3D, a larger axial range corresponds to a larger volume, and thereby to a lower density.

Alternatively, if we define density using the standard 2D definition of emitters over area, higher density correlates with a more ill-posed inverse problem regardless of the axial range. Nonetheless, the "apparent" 2D density is still PSF-dependent, and needs to be calibrated with respect to the standard microscope in-focus PSF using an appropriate conversion factor. To calibrate this PSF-dependent conversion factor, we used the following simple approach: for each axial slice within the PSF axial range, we counted the number of nonzero pixels on the CCD that are above 15% of the maximal intensity (Fig. SN1.1a). We then took the average number of nonzero pixels across the PSF axial range to be the mean lateral extent of the PSF. Finally, the resulting conversion factor is given by the mean lateral extent of the 3D PSF divided by the mean lateral extent of the standard in-focus PSF. This resulted in a conversion factor of ≈6 for the Double Helix (DH) PSF covering a 2 μm axial range, and ≈11.2 for the Tetrapod PSF covering a 4 μm axial range.

Now, to understand what density range for a given PSF qualifies as "high density", we can translate it to the standard in-focus PSF using the conversion factors above. For example, a density of 1.34 $\left[\frac{emitters}{\mu m^2}\right]$ for the standard in-focus PSF translates to a density of ≈0.22 $\left[\frac{emitters}{\mu m^2}\right]$ for the DH PSF, and a density of ≈0.12 $\left[\frac{emitters}{\mu m^2}\right]$ for the Tetrapod PSF. Assuming the emitters are uniformly spread across the middle portion of a $\approx 13 \times 13 \mu m^2$ FOV, this corresponds to imaging 238 emitters with the standard in-focus PSF, as opposed to 40 emitters with the DH PSF, and only 21 emitters with the Tetrapod PSF, all of which will result in ≈4100 nonzero pixels on the sensor (Fig. SN1.1b). This also explains why the highest density with the DH PSF in the EPFL 3D challenge [4] was ≈0.3 $\left[\frac{emitters}{\mu m^2}\right]$.

Therefore, due to the considerations explained above and following [4] we defined density throughout this work as the number of emitters divided by the FOV. In addition, example PSF images were included when necessary to provide visual guidance. However, if the reader is interested in translating these numbers into the standard volumetric density ($\left[\frac{emitters}{\mu m^3}\right]$), he can simply multiply the reported density by a factor of $\frac{1}{4}$ as our axial range throughout this work is 4 μm.

Of course, it is possible to use more complicated metrics that quantify the lateral spread of the PSF, but here we choose this simple approach as it is intuitive and easily relates the 3D case to the more familiar 2D counterpart where the definition of "high density" is more established.

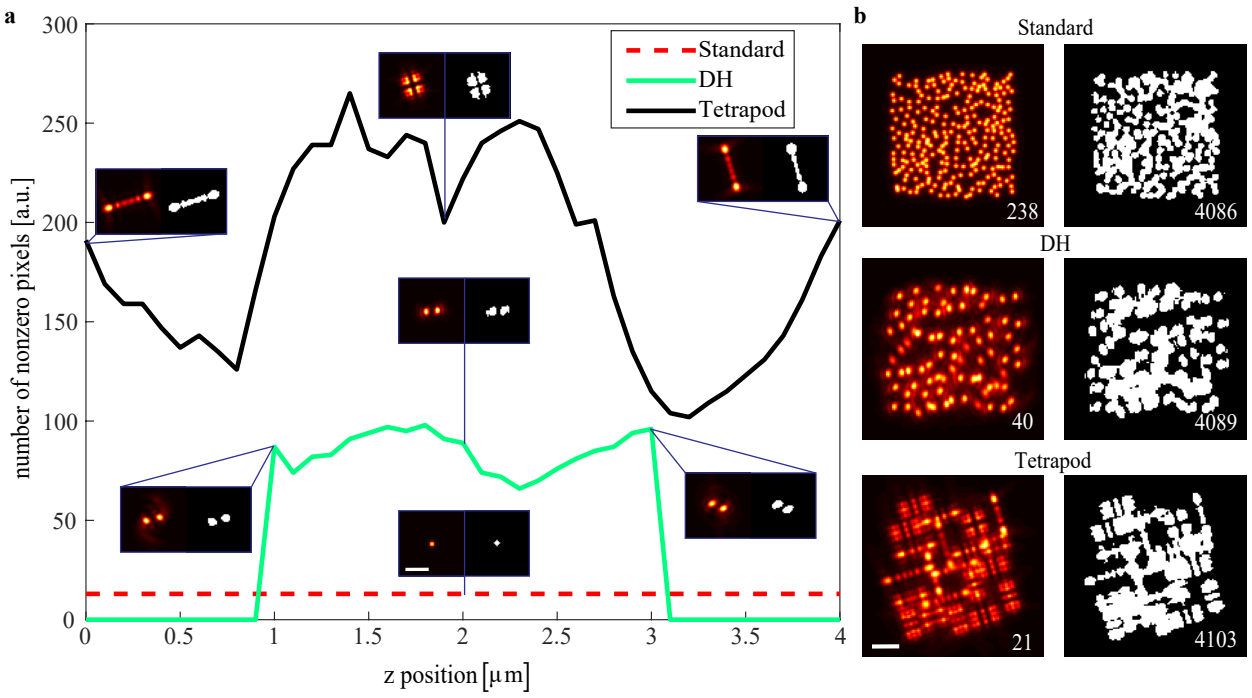


Fig. SN1.1. Apparent density. **a** Number of nonzero pixels on the CCD throughout the axial range for the standard in-focus PSF (dashed red), the DH PSF (greed) and the Tetrapod PSF (black). Insets show examples PSFs and their thresholded version. **b** Example FOV (left) and its corresponding thresholded version (right) with the standard in-focus PSF (top row), the DH PSF (middle row), and the Tetrapod PSF (bottom row). The number of emitters, and the number of nonzero pixels is stated in the lower right corner of the appropriate image. Scale bars are 2 μm.

Finally, note that all of the above discussion disregards local density, and define density as a global measure over the entire FOV. Although, to truely understand the difficulty of localizing nearby emitters one needs to take into account their clustering in space (e.g. a similar analysis to [5]). For example, in SMLM experiments emitters are naturally bound to the underlying 3D structures they are labeling; they are not uniformly distributed in the FOV. Therefore, locally, one can measure highly overlapping PSFs (Fig. 3a main text), even when the average density is low. However, this makes the comparison of different PSFs more involved, and is beyond the scope of this work.

2 CNN Architectures

Due to considerations explained later on, we used two different CNN architectures for localizing emitters and for learning a phase mask. First, let us discuss the rationale behind the localization architecture (Fig. SN2.1). As a general rule of thumb, we tried designing simple architectures with the minimal number of parameters needed to solve the problem. Moreover, to handle arbitrary image dimensions we used fully-convolutional CNNs [6]. Furthermore, since the input image contains rich information that needs to be carefully decoded, we passed it via concatenation to all consecutive layers with similar dimensions as an additional feature. To prevent these connections from making the network extremely sensitive to the normalization scheme of the input image, we added a Batch Normalization (BN) layer [7] at the beginning of the architecture that acts as a regularizer and can learn the right normalization of the input image from our training set. To benefit from the input image statistics at test time, we first alter its mean and standard deviation such that it matches the training set statistics:

$$I_{in} = \left(\frac{I_{test} - \mu_{test}}{\sigma_{test}} \right) \times \sigma_{train} + \mu_{train} \tag{S1}$$

Where $\mu_{train}, \mu_{test}, \sigma_{train}, \sigma_{test}$ are the mean and standard deviation of the pixel values of the training set images, and the test image respectively. Then, we feed I_{in} to the recovery net. While this is a sub-optimal normalization scheme, the resulting architectures were more robust than using an Instance Normalization [8] approach since the test image statistics can vary significantly between experiments. This normalization strategy was particularly useful for the telomere data where the SNR varied significantly between experiments, and was less important for the mitochondria data which exhibited a very similar SNR throughout the STORM experiment. Next, let us discuss the localization architecture in more details.

2.1 Localization CNN

The proposed architecture (Fig. SN2.1) has only $\approx 436K \backslash 612K$ trainable parameters and is composed of 3 main modules:

1. Multi-scale context aggregation module: we used dilated convolutions [9] to increase the receptive field of each layer while keeping a fixed number of 64 channels. We set the number of convolution blocks to $i_{max} = 5$. The maximal dilation rate d_{max} was set according to the PSF lateral footprint: $d_{max} = 16 \backslash 4$ for the Tetrapod and the learned PSF respectively (see Fig. SN2.1b). We also include skip connections to improve gradient flow [10]. Note that this is different from typical architectures used for similar localization tasks in computer vision such as 3D human pose estimation (HPE) [11, 12]. The rationale behind using a simpler architecture with far fewer parameters is that our images have an "easier" context as opposed to extreme semantic variations encountered in HPE.

2. Upsampling module: we used a simple upsampling module composed of two consecutive $\times 2$ resize-convolutions [13] to increase the lateral resolution by a factor of 4. We used nearest-neighbor interpolation to resize the images. Although more sophisticated upsampling layers with more representation capacity could be used, for example transposed convolution [14–16] or the more recent sub-pixel convolution [17], these layers require a proper initialization to avoid chekcerboard artifacts [13, 18] and are not necessary for our task. Assuming a CCD pixel-size of 110 nm, the lateral pixel-size of the upsampled features is 27.5 nm.

3. Prediction module: after super-resolving emitters in the lateral dimension, we further refine their axial position through 3 additional convolutional blocks with an increased number of channels. For a 4 μm range, we use 80/120 channels for the telomere/mitochondria samples respectively, i.e. a voxel-size of 33/50 nm in z. The final prediction is given by a 1×1 convolution followed by an element-wise HardTanh [19] to limit the output range to $[0, W]$. As there are only few emitters in a large vacancy volume the classes are highly imbalanced. To take this into account, we weight the ground truth locations by a factor of W=800 determined empirically, and we allow the output of the net to be in the range $[0, W]$. This strategy allows us to avoid gradient clipping, and enable meaningful gradients to flow throughout the network during training.

Note that depth is exchanged with channels as our architecture is composed of solely 2D convolutional layers. Afterwards, these dimensions are permuted in the recovered volume. Finally, we threshold voxel-values and find local maxima in clustered components to compile a list of 3D localizations (details in section 5).

In addition, we chose to work with a net that outputs a super-resolved volume (see section 4.2). However, recovering a vacancy grid is not truly a limitation as it can be combined with a second coordinate-regression net that outputs a continuous list of localizations [20]. Moreover, our recovery voxel-size with a 110 nm CCD pixel is either $27.5 \times 27.5 \times 33\ nm^3$ or $27.5 \times 27.5 \times 50\ nm^3$, which means assuming the net predicts the right voxel, our precision is limited at worst to ≈ 20 nm in the lateral dimension, and $\approx 17 \backslash 25$ nm in the axial dimension. This limit is achieved only when encountering an emitter near one of the voxel vertices which is a very unlikely event assuming a uniform distribution. Moreover, as confirmed by our simulations, for images with more than a single emitter the localization precision is not limited by the recovery voxel-size, especially for higher emitter densities.

Finally, the choice of the recovery voxel-size in the axial dimension ($\Delta_z = 33\backslash 50$nm) is merely computational, depending on the available GPU memory and the desired accuracy. For the telomere samples, the experimental ground truth positions were difficult to estimate precisely (see section 12), therefore we didn't bother with a smaller voxel-size. On the other hand, for the mitochondria sample we expected to recover the hollow tubular structures of the mitochondria, therefore we choose the minimal voxel-size that will still fit the entire model on GPU throughout training.

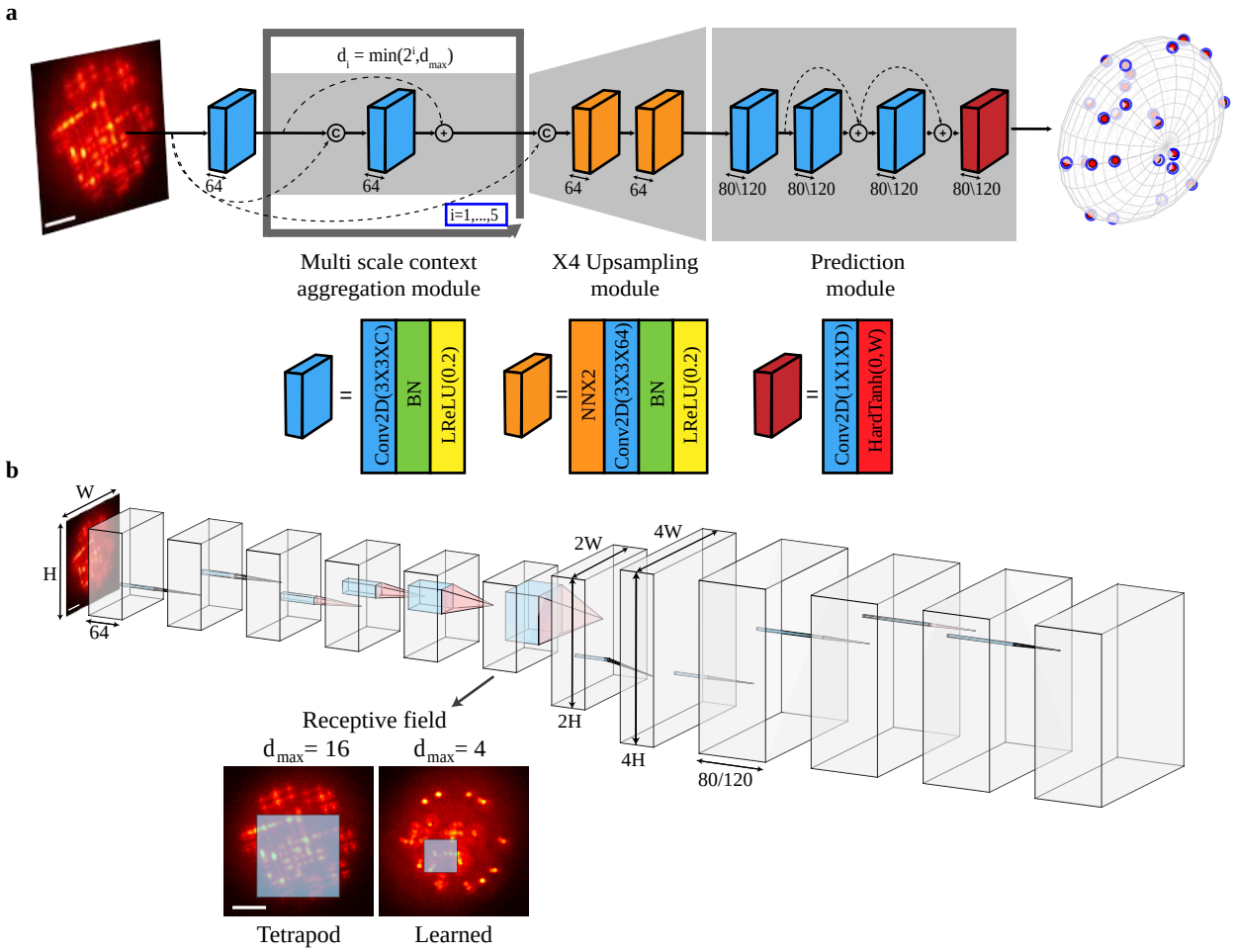


Fig. SN2.1. Localization architecture. **a** The low-resolution 2D input image I_{in} is first passed through a BN layer to normalize pixel values. Next, the normalized image I_{norm} is passed through the fully convolutional architecture where C denotes concatenation and + denotes element-wise addition. The spatial supports of all convolutional filters are 3×3. The number of channels is fixed to 64 in both the multi-scale context aggregation, and the upsampling modules. Then, the number is increased to $80\backslash 120$ for the refinement module. The prediction is given by a 1×1 convolution followed by a HardTanh activation limiting the range to $[0, \mathrm{W}]$. The output 3D high-resolution volume is translated to a list of 3D localizations through simple post-processing. An example pair of simulated-input and output are presented before and after the architecture respectively. Blue empty spheres denote simulated positions along the surface of an ellipsoid. Red spheres denote CNN detections. **b** Feature maps dimensions depicted with [21] to reflect the operation of each module. Note that in the context aggregation module the spatial support of all convolutional filters is 3×3, although their receptive field grows exponentially with the dilation rate. Blue square depicts the final receptive field for both choices of d_{max}. Scale bars are 3 μm.

2.2 Optical design CNN

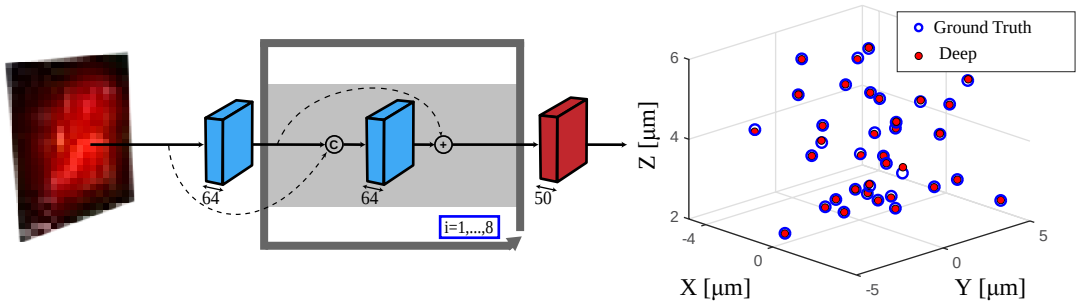


Fig. SN2.2. Phase mask learning architecture. The low-resolution 2D input image I_{in} is first passed through a BN layer to normalize pixel values. Next, the normalized image I_{norm} is passed through the fully convolutional architecture where C denotes concatenation and + denotes element-wise addition. The spatial supports of all convolutional filters are 3×3. The number of channels is fixed to 64 up until the final prediction where it is reduced to 50. The prediction is given by a 1×1 convolution followed by a HardTanh activation limiting the range to $[0, W]$. The output 3D high-resolution volume is translated to a list of 3D localizations through simple post-processing. Blue empty spheres denote simulated GT positions. Red spheres denote CNN detections. Scale bar is 3 μm.

Optimally, the architecture used for learning a phase mask should be the same architecture used for localization. Although, calculating the gradients with respect to the phase mask involve computing several FFTs in each forward and backward pass through the net. This added complexity made learning computationally inefficient, and led to inferior results. Hence, to design a phase mask we introduced several modifications to the architecture (Fig. SN2.2). First, the maximal dilation rate was set to $d_{max} = 1$, and the number of convolutional blocks was increased to $i_{max} = 8$. The receptive field after this modification is 19×19. Next, the upsampling module is eliminated and the lateral dimensions were kept similar to the input CCD image. Finally, the refinement module was also discarded, keeping only the last prediction block (Fig. SN2.1 red block) with a weighting factor of $W = 100$ and discretization of D=50 in z, resulting in an ≈isotropic voxel-size of $110 \times 110 \times 100\ nm^3$. The resulting number of trainable parameters in this modified architecture was only ≈300K.

As was noted in previous work on PSF engineering [2], we empirically observed that it is sufficient to optimize the phase mask with steps of 100 nm in the axial direction (Fig. SN2.3). Moreover, due to refractive index-mismatch, an axial shift of the emitter position is not interchangeable with a shift of the focal plane (see section 3.1). In the telomere samples we imaged, the emitters were confined to a 4 μm axial range, with the lowest being shifted $\approx 1-3$ microns from the coverslip. To account for the axial range shrinkage, we designed a PSF spanning a larger axial range of $[0, 5]$ μm with the focal plane centered in 2.5 μm. Finally, as we are first to consider engineering a microscope PSF for high-density localization, we initialized the optimization process with a phase mask implementing zero modulation, meaning, the standard microscope PSF (Fig. 1b main text).

Importantly, in contrast to previous works designing phase masks [1, 2, 22–24], we do not constrain our design space to be spanned by a fixed set of polynomials (e.g. Gauss-Laguerre modes [1], Zernike modes [2, 23, 24] or concentric rings [22]). Instead, we optimize the phase at each one of the phase mask pixels separately, since this is a much richer class of hypothesis as verified by the learned mask.

Interestingly, learning the phase mask is composed of two main phases; First, the PSF is shaped in the middle 2 μm range around the focus. Afterwards, once the localization CNN learns to correctly localize emitters in this reduced range, the mask is refined to prevent signal loss at the edges of the axial range and boost the performance at the remaining 2 μm (see Supplementary Video 4).

Of course, all of the choices above affected the learned phase mask. To study the contribution of the individual choices, we performed the following numerical experiments:

- We learned the phase mask with the localization architecture to study the effect of the net architecture on the result (Fig. SN2.3). Both architectures resulted in extremely similar PSFs regardless of the lateral pixel size in the localization architecture. However, the modified architecture provided denser gradients and distributed the photons more uniformly throughout the axial range.

- We initialized the phase mask to be the Tetrapod mask in order to start from an approximately even distribution of the photons throughout the axial range, and studied the effect of the axial design range and the localization architecture's receptive field controlled by the maximal dilation rate (Fig. SN2.4). We observed two key results in this experiment. First, with a large enough receptive field ($d_{max} = 16$) the phase mask is hardly changed regardless of the axial design range. Second, with a smaller receptive field ($d_{max} = 4$), the resulting PSF had a significantly smaller lateral footprint. This result highlights the importance of the net receptive field when the PSF is initialized to have a large lateral extent. In contrast, when we start from the standard PSF, the receptive field in both cases ($d_{max} = 4 \backslash 16$) captures the entire initial PSF, and therefore has negligible effect on the learned PSF spatial extent. Moreover, the learned phase mask using the axial range $[0, 5]$ resulted in a more uniform distribution of the photons throughout the PSF, on the expense of a slight increase in the CRLB.

- The maximal dilation rate was set to $d_{max} = 4$, the axial design range was set to $[0, 5]$, and the phase mask was initialized to the double helix (DH) mask [1] (Fig. SN2.5). First, note that the result aligns with the previous experiment emphasizing the fact

that when the PSF is initialized to have a smaller lateral footprint than the net receptive field, then it is hardly modified. This is evident in the result, as the PSF is hardly changed in the middle portion of the axial range. More interestingly, the result suggests that using our method we can extend the DH PSF to a larger axial range of 4 μm by only modifying it at the edges of the axial range.

Finally, an interesting question arises with respect to the proposed co-design approach. That is, what is the optimal PSF for a single emitter using our method? To answer this question we constrained the number of emitters in each training example to be 1, and set the maximal dilation rate to $d_{max} = 16$ in order to enable the learned PSF to have a large spatial footprint (Fig. SN2.6). The axial design range was set to $[0, 5]$, and the phase mask was initialized either to the Tetrapod mask or to zero-modulation.

Not surprisingly, when we initialized with the Tetrapod mask, the net hardly changed the phase mask. On the other hand, when we initialized with zero-modulation, the resulting phase mask was extremely different from the phase mask we obtained for the high density case. This time, the net preferred the PSF to have a large spatial extent with "dilated" features in order to ease its localization. Although, when quantifying the localization results for 10K samples we found the learned PSF to be slightly inferior to the Tetrapod in axial RMSE (Fig. SN2.6). This is partially due to our localization architecture being suboptimal for single-emitter localization.

To understand the limit of the achievable performance given our localization architecture, we next calculated the theoretical bound on the RMSE in the lateral and axial dimensions given a voxel-size of $\left(\Delta_{xy} \times \Delta_{xy} \times \Delta_z\right)$. Assuming emitters are uniformly distributed in each voxel, we can calculate the mean squared error from the middle of the voxel which is the optimal recovered position by the net:

$$\begin{aligned} MSE_{xy} &= \mathbb{E}_{(X,Y)\sim\mathcal{U}\left(\left[0,\Delta_{xy}\right]\times\left[0,\Delta_{xy}\right]\right)}\left[\left(x-\frac{\Delta_{xy}}{2}\right)^2+\left(y-\frac{\Delta_{xy}}{2}\right)^2\right]=\frac{\Delta_{xy}^2}{6} \\ MSE_z &= \mathbb{E}_{Z\sim\mathcal{U}(0,\Delta_z)}\left[\left(z-\frac{\Delta_z}{2}\right)^2\right]=\frac{\Delta_z^2}{36} \end{aligned} \tag{S2}$$

Substituting our recovery voxel-size of $(27.5 \times 27.5 \times 50)\ nm^3$ we get the following lower bounds:

$$\begin{aligned} RMSE_{xy} &= \sqrt{\frac{\Delta_{xy}^2}{6}} \approx 11\ nm \\ RMSE_z &= \sqrt{\frac{\Delta_z^2}{36}} \approx 17\ nm \end{aligned} \tag{S3}$$

Therefore, with the Tetrapod PSF, we are reaching the limit of the achievable precision with our architecture (Fig. SN2.6), and the PSF cannot be improved any further for the single-emitter case. On the other hand, starting from zero-modulation we still have some room for improvement, most likely due to optimization errors. To truly optimize the single emitter case, one needs to consider a different localization architecture that outputs continuous values, which will be addressed in future work.

Finally, we note that the discussion above regarding the achievable precision disregards our final post-processing step where we take a local average around the found local maxima. While we found this in practice to improve our axial RMSE by $\approx$5 nm for all three PSFs, this step is not part of the learning process, therefore does not help when optimizing the mask for the single-emitter case.

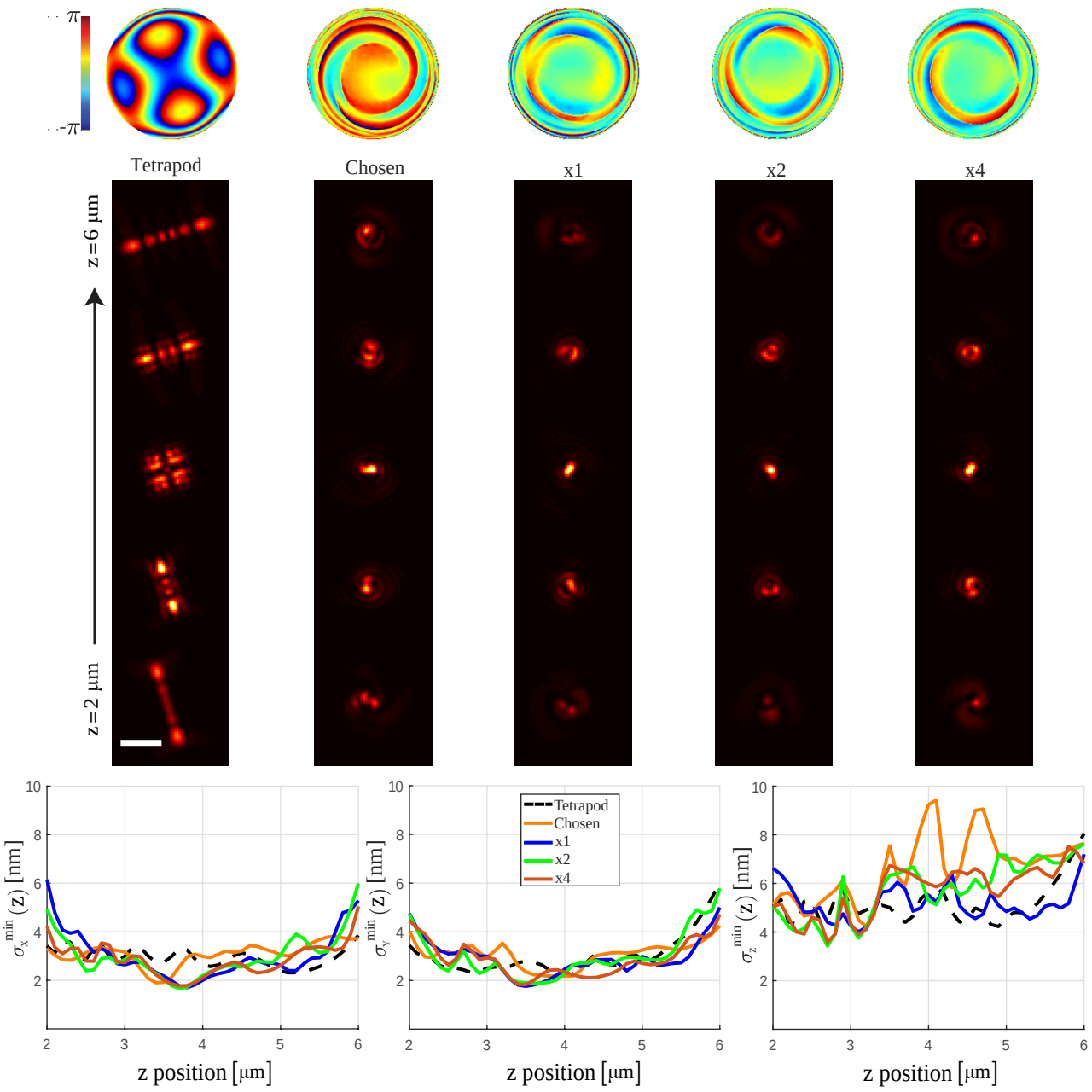


Fig. SN2.3. Effect of architecture and voxel-size on the learned PSF. We fixed the optimization and the learning hyper-parameters, set the axial range to $[2, 6]$, and learned the phase mask with the localization architecture. To test the effect of the voxel-size in xy, we tried 3 different settings: $\times 4$ - the full localization architecture ($\Delta_{xy} = 27.5$ nm, $\Delta_z = 50$ nm), $\times 2$ - the localization architecture with only one upsampling layer ($\Delta_{xy} = 55$ nm, $\Delta_z = 50$ nm), and $\times 1$ - the localization architecture without upsampling ($\Delta_{xy} = 110$ nm, $\Delta_z = 50$ nm). The learned masks were similar in all 3 cases. Moreover, compared to the learned mask with the modifications proposed in section 2.2, the resulting PSFs had a lower(better) CRLB (bottom plots) on the expense of faster signal loss at the edges of the axial range. The CRLB was calculated assuming $30K$ signal counts with 160 counts per-pixel background. Similarly to [25] differentiation is done numerically with 1 nm perturbations. Scale bar is 2 μm.

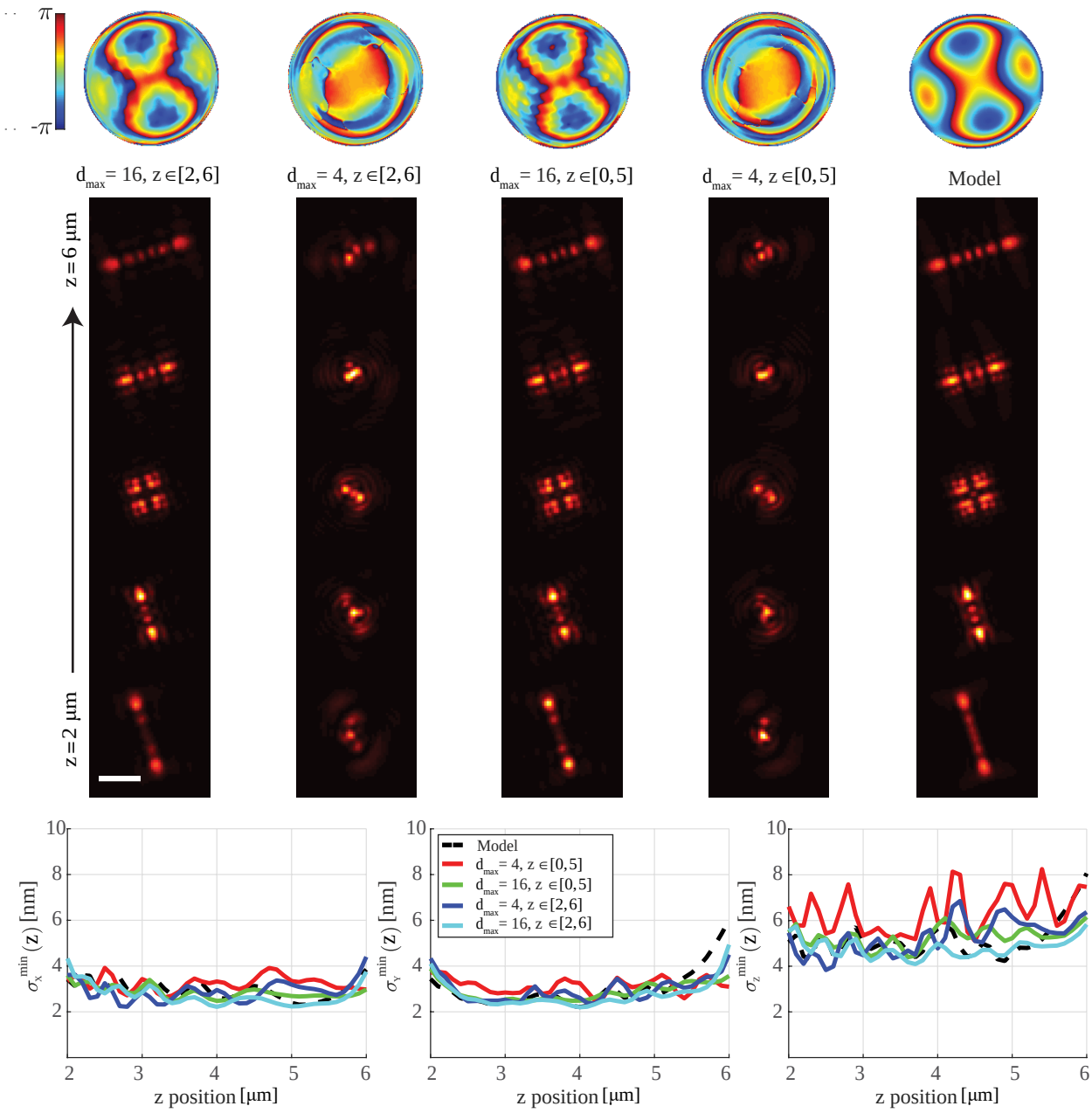


Fig. SN2.4. Effect of axial design range and the localization net receptive field. The phase mask was initialized using the Tetrapod model mask (right column), the maximal dilation rate was either $d_{max} = 4$ or $d_{max} = 16$ corresponding to a receptive field of either 65×65 or 21×21 pixels, and the axial optimization range was either $[0, 5]$ or $[2, 6]$ μm. The larger receptive field resulted in minor modifications to the PSF, while the smaller receptive field enforced it to have a smaller lateral footprint. Moreover, designing the PSF using the larger and lower axial range ($[0, 5]$ μm resulted in a more even distribution of photons on the expense of a slightly increased CRLB (bottom plots). The CRLB was calculated assuming $30K$ signal counts with 160 counts per-pixel background. Similarly to [25] differentiation is done numerically with 1 nm perturbations. Scale bar is 2 μm.

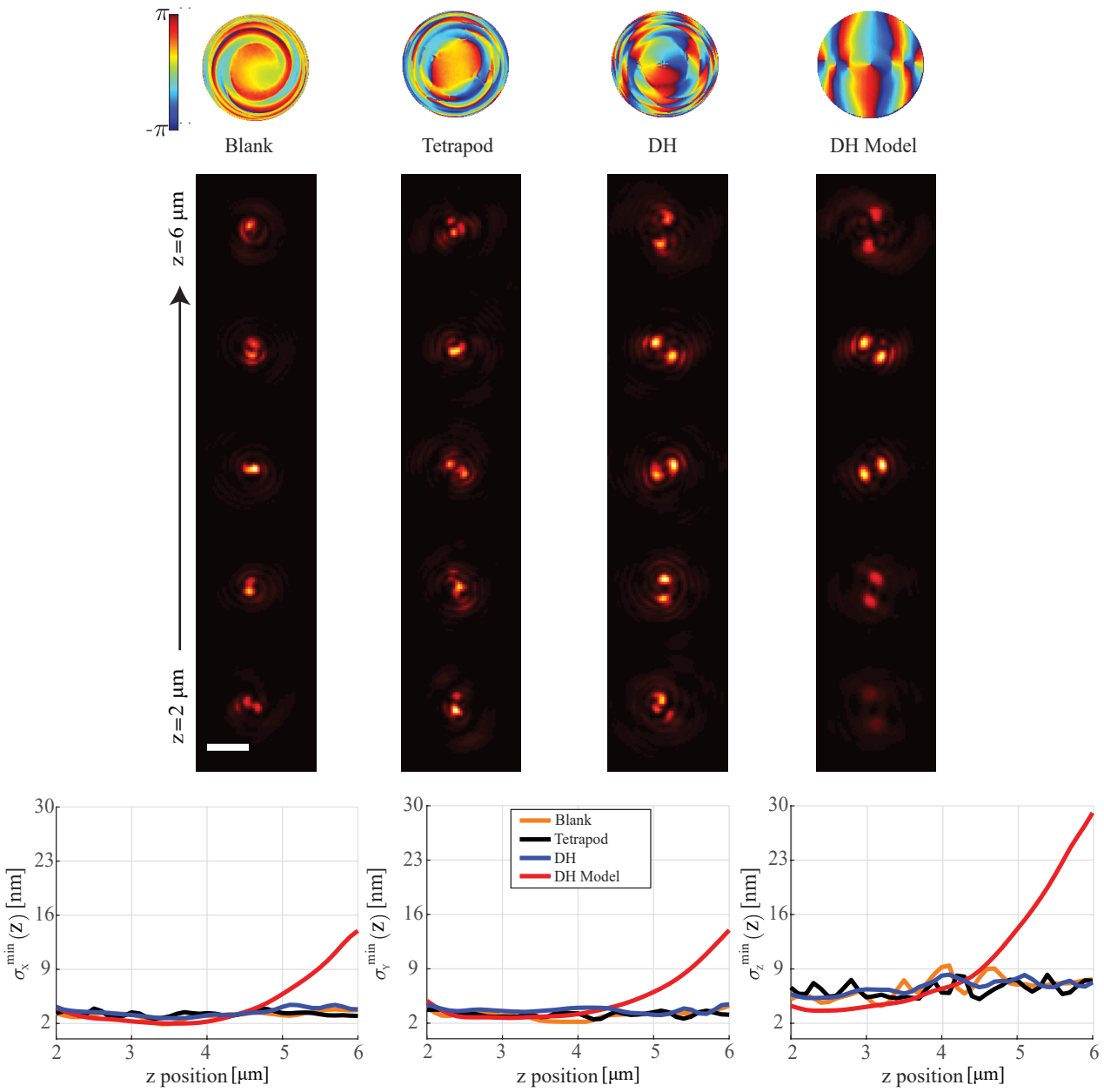


Fig. SN2.5. Double helix initialization to extend the PSF axial range. The phase mask was initialized using three different options: (1) Zero-modulation mask, (2) Tetrapod mask, and (3) Double helix mask (right column). The maximal dilation rate was $d_{max} = 4$, and the axial design range was $([0, 5]$ μm. Interestingly, the net did not modify the double helix PSF in its working range, and only modified it at the edges to capture the full 4 μms. Although compared to the PSF learned with an initial zero-modulation/Tetrapod mask, the double helix initialized PSF was larger, making it potentially harder to localize at extremely high densities. The CRLB (bottom plots) was calculated assuming $30K$ signal counts with 160 counts per-pixel background. Similarly to [25] differentiation is done numerically with 1 nm perturbations. Scale bar is 2 μm.

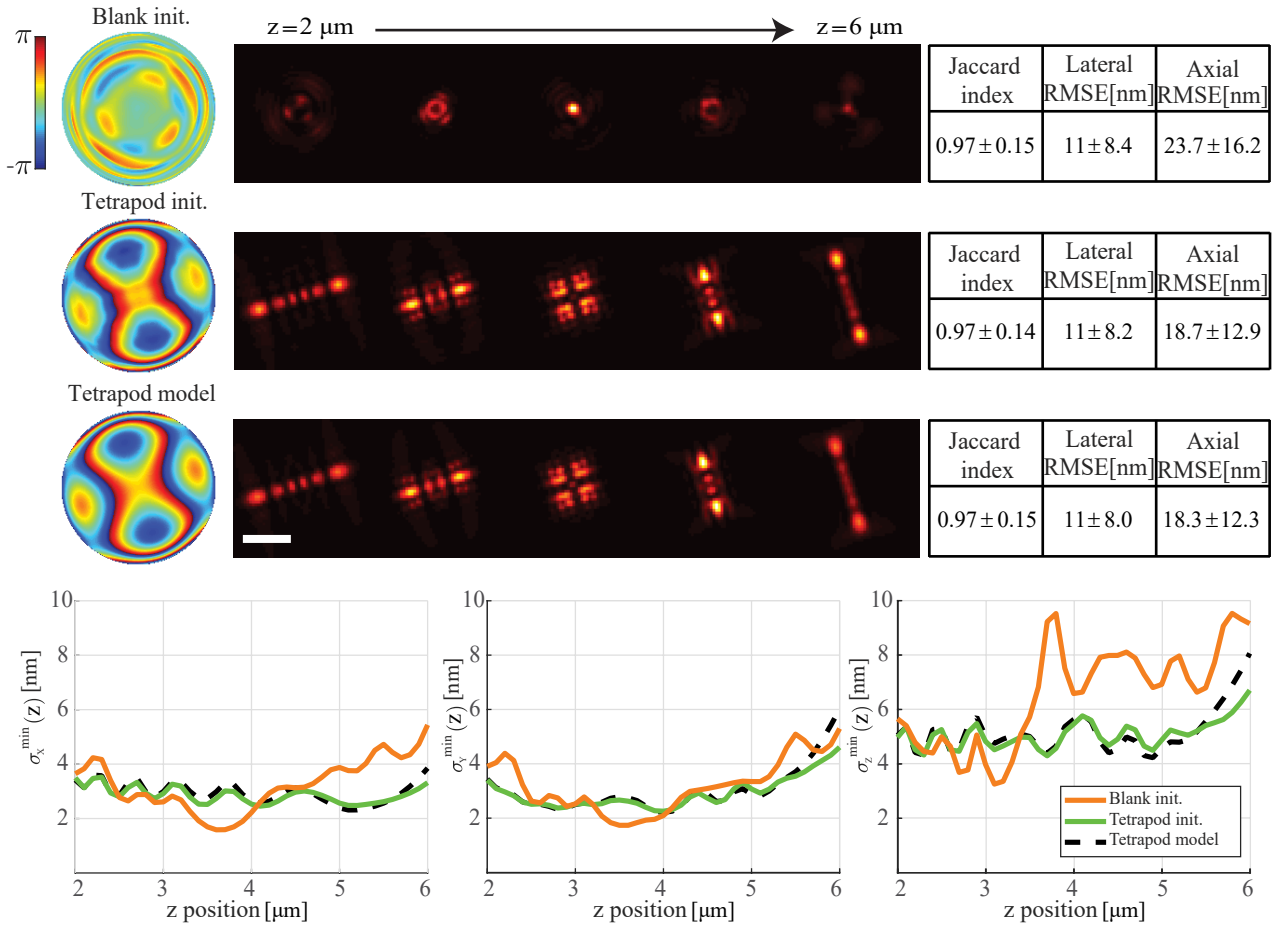


Fig. SN2.6. Single emitter phase mask learning. The number of emitters per FOV was dropped to 1. The phase mask was initialized to either blank/zero-modulation, or to the Tetrapod mask. The maximal dilation rate was $d_{max} = 16$, and the axial design range was $([0,5]\ \mu$m. The learned PSF for the zero-modulation mask had "dilated" features, while the Tetrapod PSF was hardly changed. Compared to the initial Tetrapod mask, the PSF learned for the blank initilization had slightly worse performance in axial RMSE, most likely due to optimization errors. The CRLB (bottom plots) was calculated assuming $30K$ signal counts with 160 counts per-pixel background. Similarly to [25] differentiation is done numerically with 1 nm perturbations. Scale bar is 2 μm.

3 Physical layer

3.1 Imaging model

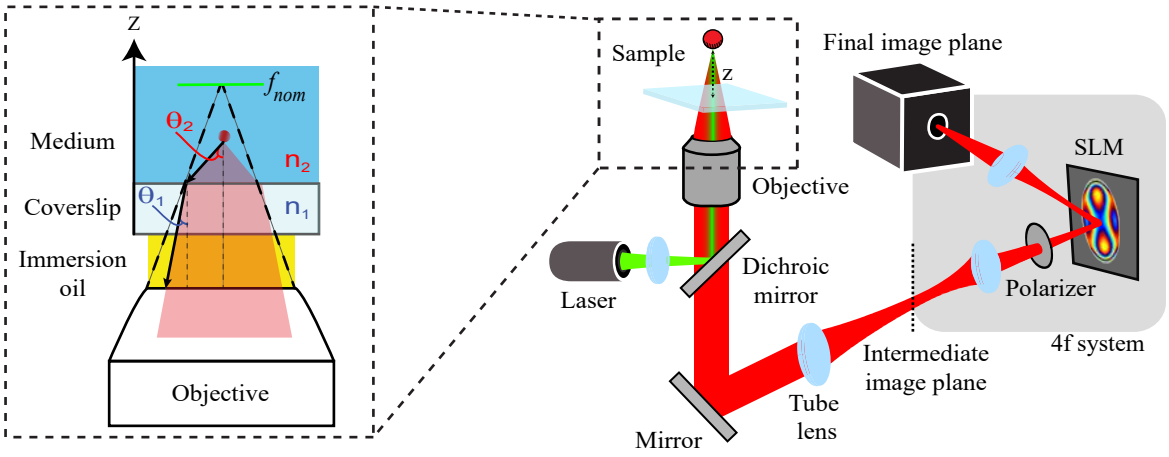


Fig. SN3.1. Imaging model. The light emitted from a fluorescent microscopic particle with distance z_0 from the coverslip propagates through the suspension medium (refractive index of water $n_2 \approx 1.334$) with an angle of θ_2, and refraction occurs at the interface between the medium and the coverslip. The refracted light propagates in glass/immersion oil (refractive index of $n_1 \approx 1.517$) with an angle θ_1 and is collected by the objective which is focused at f_{nom}.

The imaging model used in this work is based on the scalar diffraction approximation of light emitted from an isotropic fluorescent emitter [26]. The optical setup is a 4f-extended microscope with a phase mask implemented by an SLM in the back-focal plane (Fig. SN3.1; reproduced from the main text with additional details for convenience). We assume the emitter is suspended in a medium with a refractive index close to that of water $n_2 \approx 1.334$, and is imaged using an oil-immersed objective with a refractive index of $n_1 \approx 1.517$ matching the glass of the coverslip. Under these assumptions, the PSF in image plane $I_r(u,v)$ due to a point source located at $r = (x_0, y_0, z_0)$ is given by:

$$I_r(u,v) \propto \left|\mathcal{F}_{2D}\left(E_r(\rho,\phi)\right)\right|^2 \tag{S4}$$

Where $E_r(\rho,\phi)$ is the electric field at the back focal plane (BFP), and $\mathcal{F}_{2D}$ denote the two-dimensional Fourier transform. Using Abbe sine rule, the physical dimension of the limiting radius at the BFP due to our 4f-system extension is given by:

$$r_{phys} = \frac{f_{4f}\mathrm{NA}}{\sqrt{\mathrm{A_M^2} - \mathrm{NA}^2}} \tag{S5}$$

Where f_{4f} is the focal length of each lens in the 4f system, NA is the numerical aperture of the objective, and $\mathrm{A_M}$ is the magnification of the microscope. For convenience, we define two sets of coordinates in the BFP: cartesian (ζ,η), and polar (ρ,ϕ). The polar coordinates are normalized such that $\rho = 1$ at the limiting aperture given by $\frac{\mathrm{NA}}{n_1}$. As for the cartesian coordinates, they are given by:

$$\begin{aligned}\zeta &= r_{phys}\rho cos(\phi)\\ \eta &= r_{phys}\rho sin(\phi)\end{aligned} \tag{S6}$$

The intensity of light is assumed to be uniform within the aperture:

$$circ(\rho) = \begin{cases}1 & \rho \le 1\\ 0 & \text{otherwise}\end{cases} \tag{S7}$$

Next, let us derive the terms comprising the phase of $E_r(\rho,\phi)$. First, the phase induced by the phase mask M deployed on the SLM is simply given by the mask itself:

$$\Phi_{mask} = M \tag{S8}$$

Let λ denote the emission wavelength, $k_1 = \frac{2\pi n_1}{\lambda}$ denote the wave-number of the electrical in oil, $k_2 = \frac{2\pi n_2}{\lambda}$ denote the wave-number of the electrical field in water, and f_{nom} denote the nominal focal plane. For a point source located above a water-oil interface (Fig. SN3.1) the axial phase is comprised of two parts; First, the axial phase accumulated in water (suspension-medium) due to the emitter distance from the coverslip z_0:

$$\Phi_{ax_2} = z_0 k_2 \cos\theta_2 \tag{S9}$$

Second, the axial phase accumulated in oil due to the focus setting (f_{nom}) which is independent of the emitter position:

$$\Phi_{ax_1} = \left(f_{obj} - f_{nom}\right) k_1 \cos\theta_1 \tag{S10}$$

Where f_{obj} is the objective focal length. To explicitly calculate the terms in equations (S9) and (S10), we write Snell's law on the interface: $n_1 \sin\theta_1 = n_2 \sin\theta_2$, and use the trigonometric relation $\cos\theta = \sqrt{1 - \sin^2\theta}$. The resulting axial phases are given by:

$$\begin{aligned} \Phi_{ax_1} &= -f_{nom} k_1 \sqrt{1 - \rho^2} \\ \Phi_{ax_2} &= z_0 k_2 \sqrt{1 - \left(\frac{n_1}{n_2}\rho\right)^2} \end{aligned} \tag{S11}$$

Where f_{obj} was dropped since it is already corrected for by the objective. Finally, the lateral shift of the point source (x_0, y_0) is modelled using a linear phase:

$$\Phi_{lat} = 2\pi \left(x_0 \frac{\zeta \mathrm{A_M}}{\lambda f_{4f}} + y_0 \frac{\eta \mathrm{A_M}}{\lambda f_{4f}} \right) \tag{S12}$$

Hence, put equations (S8), (S11), and (S12) together we get the following imaging model:

$$\begin{aligned} I_r(u,v) &\propto \left| \mathcal{F}_{2D}\left(circ(\rho)\, e^{j\left(\Phi_{mask} + \Phi_{ax_2} + \Phi_{ax_1} + \Phi_{lat}\right)} \right) \right|^2 \\ &\propto \left| \mathcal{F}_{2D}\left(circ(\rho)\, e^{j\left(M + z_0 k_2 \sqrt{1 - \left(\frac{n_1}{n_2}\rho\right)^2} - f_{nom} k_1 \sqrt{1-\rho^2} + 2\pi\left(x_0 \frac{\zeta \mathrm{A_M}}{\lambda f_{4f}} + y_0 \frac{\eta \mathrm{A_M}}{\lambda f_{4f}} \right) \right)} \right) \right|^2 \end{aligned} \tag{S13}$$

To achieve an exact equality the resulting image in equation (S13) needs to be rescaled with the amount of signal photons $N_{photons}$. In practice, experimental data appears slightly blurred compared to equation (S13) due to finite emitter size and aberrations not captured by the model [27]. To remedy this, we blur the result of equation (S13) with a small Gaussian filter.

Note that we do not account for dipole effects [25] and instead assume isotropic emission. Moreover, the model disregards the super-critical angle fluorescence (SAF) component which is observed when imaging in small axial ranges (< 1 μm) from the coverslip [25, 28]. Finally, we also neglected the intensity apodization factor at the BFP [29]. Nonetheless, since the model was able to describe our experimental data with satisfying accuracy, we made these simplifications to reduce complexity and accelerate our computations.

In contrast to an interpolation-based approach [4, 30, 31], a pupil function approach (equation (S13)) combined with a phase retrieval procedure [32] is able to accurately model emitters that are distant from the coverslip (> μm), potentially alleviating the need for a depth-dependent calibration [33].

3.2 Poisson noise approximation

An accurate noise model for an EMCCD camera [4, 34] takes into account three major sources of stochastic noise: shot noise produced by the fluorescence background and signal, Gaussian read out noise produced by the electronics, and electron multiplication noise introduced by the gain process. Our measurements of fixed cells were taken with an sCMOS camera [35] so we did not include the electron multiplication noise. Additionally, for live MEF cells the gain noise was well approximated by a Gaussian. Therefore, in our simulations we assumed a gain of 1, and photons were equivalent to counts. Assuming we are operating in high photon counts (with no saturation), the readout noise is negligible and the dominant noise source is the Poisson shot noise. In this case, by the law of large numbers, we can approximate the Poisson noise by a Gaussian noise using the central limit theorem:

$$\begin{aligned} y &\approx Poiss\left(\lambda = I_{model} + b\right) \\ &\approx \mathcal{N}\left(\mu = I_{model} + b, \sigma^2 = I_{model} + b\right) \end{aligned} \tag{S14}$$

Where I_{model}, b are the noiseless model image and the additive background respectively. To enable differentiability of the noise sampling operation, we apply the reparametrization trick [36], and implement the Gaussian noise approximation as:

$$y \approx I_{model} + b + \sqrt{I_{model} + b} \times \epsilon, \quad \text{where} \quad \epsilon \sim \mathcal{N}(0,1) \tag{S15}$$

In the backward pass, the standard noise realization ϵ act as a constant, and hence the overall operator is differentiable:

$$\frac{\partial y}{\partial I_{model}} = 1 + \frac{1}{2\sqrt{I_{model} + b}} \times \epsilon \tag{S16}$$

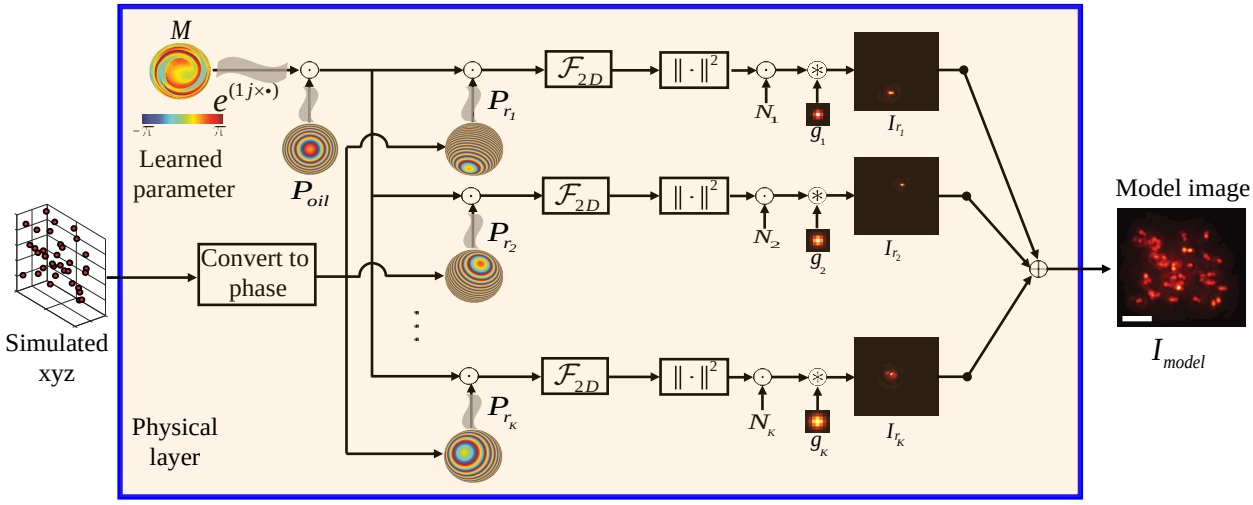


Fig. SN3.2. Physical simulation layer. The physical simulation layer is essentially the imaging model in equation (S13) viewed as a computational graph, and parametrized by the phase mask. This layer accepts simulated emitter positions as input, calculates an image per emitter, and outputs the 2D model image corresponding to the current mask. The emitters are assumed to be spatially incoherent, hence the output image is given by the incoherent sum of the individual intensity patterns. During training, in each iteration we randomly sample the number of emitters K, the number of counts per-emitter $\{N_i\}_{i=1}^{K}$, the Gaussian blur per-emitter $\{g_i\}_{i=1}^{K}$, and update the phase mask M via backpropagation. Scale bar is 3 μm.

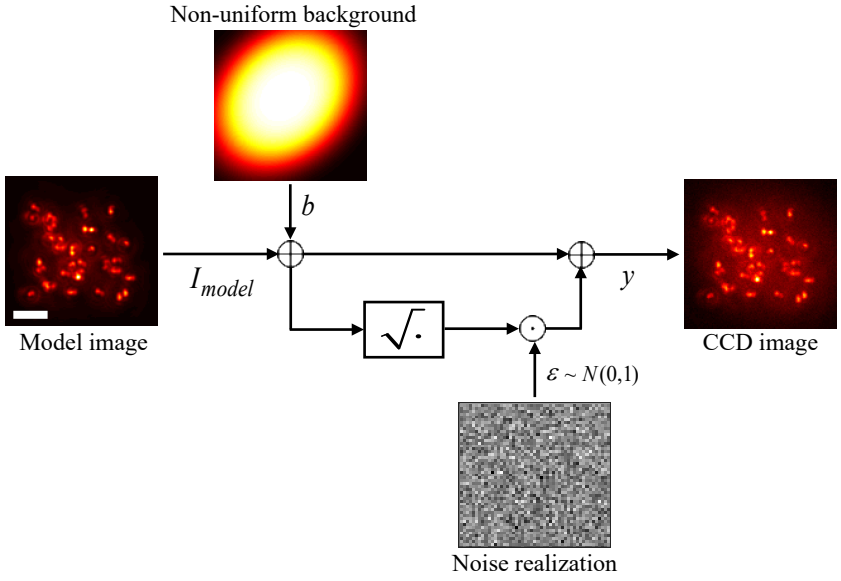


Fig. SN3.3. Noise approximation. The mean counts distribution per-pixel is given by the sum of the noiseless model image I_{model} and the non-uniform background b. Assuming Poisson statistics this is also the noise variance. Next, to implement a noise variance proportional to the mean, the sum image is passed through an element-wise squared root operation, and multiplied element-wise with a standard Gaussian noise realization. The simualted image on the CCD is modelled by the sum of the mean counts distribution and the noise approximation term. Scale bar is 3 μm.

Of course, our approach is trivially extended with an additive read-out noise realization. In fact, we needed to include this noise source in the training data for the STORM experiment. On the other hand, the telomeres data was shot-noise limited, hence there we did not bother with this extension. Note that the background term b is not limited to a constant number of counts per pixel. In fact, to empirically fit our experimental telomeres data we include a non-uniform background (Fig. SN3.3) modelled by a super-Gaussian function:

$$b = A \exp\left(-\left(\alpha_1 (x - x_0)^2 + 2\alpha_2 (x - x_0)(y - y_0) + \alpha_3 (y - y_0)^2\right)^2\right) + B \tag{S17}$$

With,

$$\alpha_1 = \frac{\cos^2\theta}{2\sigma_x^2} + \frac{\sin^2\theta}{2\sigma_y^2}, \quad \alpha_2 = -\frac{\sin 2\theta}{4\sigma_x^2} + \frac{\sin 2\theta}{4\sigma_y^2}, \quad \alpha_3 = \frac{\sin^2\theta}{2\sigma_x^2} + \frac{\cos^2\theta}{2\sigma_y^2}$$

Where B is a baseline value, A is a normalizing constant, (x_0, y_0) is the centroid, σ_x, σ_y are the on-axis standard deviations, and θ is the blob angle. On the other hand, for the STORM experiment we got rid of the non-uniform background by simply subtracting the minimum value per-pixel over the entire acquired stack. This emphasizes an inherent advantage of neural nets over most existing localization methods: tremendous flexibility to cope with a variety of observed challenges.

3.3 Gradient calculation

To compute the gradient of a scalar real-valued function ℓ of a complex-valued variable z, we can treat the real and imaginary parts of z as free variables and compute the gradient of ℓ with respect to each of them individually. This can be done most conveniently through the following formalism.

For a scalar real-valued function ℓ of a complex-valued variable z, the gradient is defined as [37, 38]:

$$\nabla_\ell(z) = \frac{\partial \ell}{\partial \operatorname{Re}(z)} + j\frac{\partial \ell}{\partial \operatorname{Im}(z)} \tag{S18}$$

Moreover, since our graph of mathematical expressions include complex-valued intermediate variables, the usual chain rule cannot be applied. Instead, for $f(z) = u + jv$ and $g(z) = r + js$, the gradient of the real-valued function ℓ with respect to their composition $f \circ g$ is computed via the "generalized chain rule" (GCR) [37, 38]:

$$\nabla_\ell(g) = \operatorname{Re}(\nabla_\ell(f))\left(\frac{\partial u}{\partial r} + j\frac{\partial u}{\partial s}\right) + \operatorname{Im}(\nabla_\ell(f))\left(\frac{\partial v}{\partial r} + j\frac{\partial v}{\partial s}\right) \tag{S19}$$

For a thorough and detailed analysis of the complex gradient operator the reader is referred to [39].

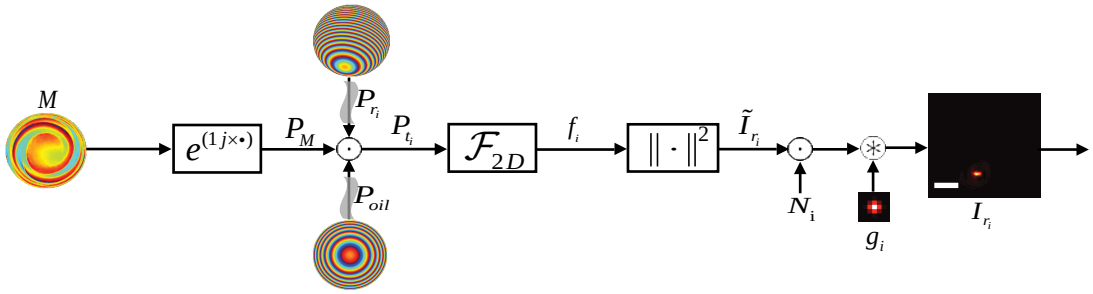


Fig. SN3.4. Single-emitter image generation pipeline. The physical simulation layer graph of operations is composed of K parallel single-emitter image generation pipelines, where the differences between different pipelines are the phase due to the emitter position P_{r_i}, the number of signal counts N_i, and the emitter size accounted for by a Gaussian blur g_i. Scale bar is 3 μm.

Next, to optimize the phase mask in the physical layer (Fig. SN3.2), we need to compute the gradient of our real-valued loss function ℓ with respect to the phase mask. We will not dwell on the gradients of ℓ with respect to the CNN parameters as this is taken care of by the automatic differentiation framework [40]. Instead, we assume we are given the gradient of ℓ with respect to the physical layer output which is the noiseless model image I_{model} (Fig. SN3.2).

When applying the back-propagation algorithm through a computational graph, a summation is replaced with a fork, and a fork is replaced with summation. Moreover, note that if we shift the global phase term accounting for phase accumulated in oil into each "single-emitter image generation pipeline" (Fig. SN3.4), the gradient back-propagated through each such pipeline will admit a similar expression up to a different position-induced phase term accounting for phase accumulated in water. Hence, for simplicity, we derive the gradient of a single pipeline while keeping in mind that the final gradient will be given by a summation of gradients over all pipelines.

Given the gradient of the loss with respect to the emitter final image $\frac{\partial \ell}{\partial I_{r_i}} = \frac{\partial \ell}{\partial I_{model}}$, the gradient $\frac{\partial \ell}{\partial \tilde{I}_{r_i}}$ is given by:

$$\frac{\partial \ell}{\partial \tilde{I}_{r_i}} = N_i \odot g_i \circledast \frac{\partial \ell}{\partial I_{r_i}} \tag{S20}$$

Where $\circledast$ denotes convolution, $\odot$ denotes a Hadamard product, and g_i is not transposed since it is a symmetric Gaussian filter. Next, applying the definition in equation (S18) to the relation $\tilde{I}_{r_i} = \|f_i\|^2$ we get:

$$\begin{aligned} \frac{\partial \tilde{I}_{r_i}}{\partial f_i} &= \frac{\partial \tilde{I}_{r_i}}{\partial \operatorname{Re}(f_i)} + j \times \frac{\partial \tilde{I}_{r_i}}{\partial \operatorname{Im}(f_i)} = \\ &= 2\operatorname{Re}(f_i) + j \times 2\operatorname{Im}(f_i) = 2f_i \end{aligned} \tag{S21}$$

Furthermore, since the Discrete Fourier Transform (DFT) is a linear operator its gradient is simply the transformation matrix itself. During backpropagation, this gradient is conjugated, hence, by DFT unitarity, this corresponds to the application of the inverse transform [41]:

$$\begin{aligned} \frac{\partial \ell}{\partial P_{t_i}} &= \mathcal{F}_{2D}^{-1}\left\{\frac{\partial \ell}{\partial f_i}\right\} = \\ &= \mathcal{F}_{2D}^{-1}\left\{\frac{\partial \ell}{\partial \tilde{I}_{r_i}} \odot \frac{\partial \tilde{I}_{r_i}}{\partial f_i}\right\} = \\ &= \mathcal{F}_{2D}^{-1}\left\{N_i \odot g_i \circledast \frac{\partial \ell}{\partial I_{r_i}} \odot 2f_i\right\} \end{aligned} \tag{S22}$$

Let $P_{n_i} = P_{r_i} \odot P_{oil}$ denote the combined phase acculumated in water and oil for emitter i. We apply the definition in equation (S19) to compute the gradient with respect to P_{M_i}:

$$\begin{aligned} \frac{\partial \ell}{\partial P_{M_i}} = &\operatorname{Re}\left(\frac{\partial \ell}{\partial P_{t_i}}\right) \odot \left(\frac{\partial \operatorname{Re}(P_{t_i})}{\partial \operatorname{Re}(P_{M_i})} + j \times \frac{\partial \operatorname{Re}(P_{t_i})}{\partial \operatorname{Im}(P_{M_i})}\right) + \\ &\operatorname{Im}\left(\frac{\partial \ell}{\partial P_{t_i}}\right) \odot \left(\frac{\partial \operatorname{Im}(P_{t_i})}{\partial \operatorname{Re}(P_{M_i})} + j \times \frac{\partial \operatorname{Im}(P_{t_i})}{\partial \operatorname{Im}(P_{M_i})}\right) \end{aligned} \tag{S23}$$

Substituting $P_{t_i} = P_{n_i} \odot P_{M_i}$ in equation (S23) we get:

$$\begin{aligned} \frac{\partial \ell}{\partial P_{M_i}} = &\operatorname{Re}\left(\frac{\partial \ell}{\partial P_{t_i}}\right) \odot (\operatorname{Re}(P_{n_i}) - j \times \operatorname{Im}(P_{n_i})) + \\ &\operatorname{Im}\left(\frac{\partial \ell}{\partial P_{t_i}}\right) \odot (\operatorname{Im}(P_{n_i}) + j \times \operatorname{Re}(P_{n_i})) \end{aligned} \tag{S24}$$

Once more, we apply the definition in equation (S19) again to compute the gradient with respect to M_i:

$$\frac{\partial \ell}{\partial M_i} = \operatorname{Re}\left(\frac{\partial \ell}{\partial P_{M_i}}\right) \odot (-sin(M_i)) + \operatorname{Im}\left(\frac{\partial \ell}{\partial P_{M_i}}\right) \odot cos(M_i) \tag{S25}$$

Note that M is replicated to all pipelines, hence $M_i = M,\ \forall i \in \{1, ..., K\}$. Although, we keep the index i to denote the gradient of ℓ with respect to M back-propagated through pipeline i. Now, recall that a fork in the forward pass of a computational graph is replaced with summation in the backward pass. Hence, the final gradient with respect to M is the sum of all gradients $\frac{\partial \ell}{\partial M_i},\ \forall i \in \{1, ..., K\}$. The steps for calculating $\frac{\partial \ell}{\partial M}$ are summarized in algorithm 1. This gradient was validated numerically using autograd *gradcheck* function.

Algorithm 1: Calculation of $\frac{\partial \ell}{\partial M}$

Input : $M, \{P_{n_i}, f_i, g_i, N_i\}_{i=1}^{K}, \frac{\partial \ell}{\partial I_{model}}$
Output: $\frac{\partial \ell}{\partial M}$
for $i \leftarrow 1$ *to* K **do**
  $\frac{\partial \ell}{\partial P_{t_i}} \leftarrow \mathcal{F}_{2D}^{-1}\left\{N_i \odot g_i \circledast \frac{\partial \ell}{\partial I_{model}} \odot 2f_i\right\}$
  $\frac{\partial \ell}{\partial P_{M_i}} \leftarrow \operatorname{Re}\left(\frac{\partial \ell}{\partial P_{t_i}}\right) \odot (\operatorname{Re}(P_{n_i}) - j \times \operatorname{Im}(P_{n_i})) + \operatorname{Im}\left(\frac{\partial \ell}{\partial P_{t_i}}\right) \odot (\operatorname{Im}(P_{n_i}) + j \times \operatorname{Re}(P_{n_i}))$
  $\frac{\partial \ell}{\partial M_i} \leftarrow \operatorname{Re}\left(\frac{\partial \ell}{\partial P_{M_i}}\right) \odot (-sin(M_i)) + \operatorname{Im}\left(\frac{\partial \ell}{\partial P_{M_i}}\right) \odot cos(M_i)$
end
return $\frac{\partial \ell}{\partial M} = \sum_{i=1}^{K} \frac{\partial \ell}{\partial M_i}$

4 Training details

4.1 Training set

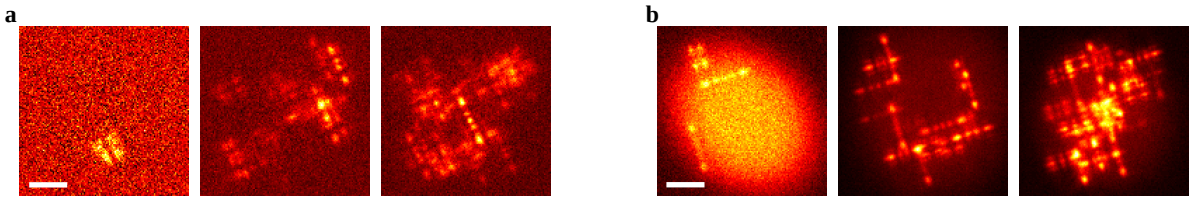


Fig. SN4.1. Training examples. **a** The mitchondria training set includes read noise, and signal counts are Gamma distributed. **b** The telomeres training set includes a non-uniform background, and is composed of examples with variable emitter size (blur). Both datasets include variable emitter density and emitter signal-to-noise ratio, with the mitochondria training set (a) having significantly lower SNR. Scale bar is 3 μm.

To learn a localization CNN solely with a predefined phase mask, we simulate a training set composed of 10K simulated images and their corresponding labels which are lists of emitter positions. 9K examples were used for training with 1K examples held out for validation. Alternatively, to jointly learn the phase mask and the localization CNN parameters, the training set is composed of solely simulated emitter positions, as the respective images are being changed throughout iterations according to the phase mask.

Given a set of 3D locations, the expected model image is simulated using a pupil function approach as explained in section 3. Using a pupil function is prefered over image space interpolation methods as it can accurately capture saddle differences of the PSF. Moreover, from a computational point of view, it is preferred over a convolution followed by a down-sampling approach [4] since we can simulate emitter locations continuously and more efficiently using FFTs. Importantly, while image-space interpolation methods employing splines [30, 31, 42] can capture aberrations which are not well described by a combination of Zernike modes [43], these methods are not suitable for imaging emitters with a large axial shift, as the PSF calibration using beads on a coverslip will not accurately describe the observed PSF due to refractive index mismatch. Therefore, this flexibility in the pupil-function approach is of critical when imaging in cells.

To accurately model experimental data in our simulations we followed the approach of [44] to retrieve the aberrated pupil function for the Tetrapod PSF (see section 11). As for the designed PSF, we observed that the phase retreival algorithm failed to recover a reasonable aberration, and hence we generated the training set according to the model pupil function. Interestingly, most of the aberrations in Fourier plane were a result of imperfect implementation of the phase mask on the SLM given a finite set of voltages, rather than misalignment of the optical system, for example. Hence, since the aberration is mask-specific and expected to behave similarly as the implemented phase mask, it is reasonable that it was not well expressed as a linear combination of zernike modes. Although pixel-wise phase retreival algorithms can be deployed (e.g. [45, 46]), the results with the model were already pleasing, and far better compared to the Tetrapod PSF because of its suitability for high density 3D localization.

To make our simulations more realistic we include experimental variability in our training set. For example, for the telomeres training sets we add a non-uniform background component that is modelled by a super-Gaussian (see section 3.2) with a randomized angle in the range $\left[\frac{-\pi}{4}, \frac{\pi}{4}\right]$ [rad], randomized standard deviations in the range $\left[\frac{\text{FOV}}{5}, \frac{2\times\text{FOV}}{5}\right]$ [px], and randomized amplitudes with a baseline value in the range $[20, 30]$ [counts], and a maximal value in the range $[120, 180]$ [counts]. Furthermore, we take into account variations in particle size by convolving each sources' image with a Gaussian blur of a randomized standard deviation in the range $[0.75, 1.25]$ [px] (Fig. SN3.2). Moreover, to enforce robustness to a wide range of conditions, the density of the emitters was varied in the range $\left[\frac{1}{\text{FOV}}, \frac{35}{\text{FOV}}\right]$ $\left[\frac{\text{emitters}}{\mu m^2}\right]$ with a field-of-view (FOV) of 13×13 μm^2. Finally, to prevent the net from over-fitting intensity, the number of signal counts per emitter was varied in the range $[9K, 60K]$ [counts]. Conveniently, the simulated training set can easily incorporate additional experimental challenges such as motion blur, laser fringes, etc. This flexibility is key to making the method versatile and readily extendable for different applications.

In fact, for the STORM experiment, we found that the additive non-uniform background was not necessary since subtracting the minimal value per-pixel of the stack eliminated this issue. However, there we observed a different set of challenges. For instance, while the mean background was relatively constant throughout the FOV, the standard deviation of the read-noise was still higher in the middle of the FOV. Therefore, to take this into account we used the same super-Gaussian from before to scale the standard deviation of the read noise in the range $[8, 12]$ [counts]. In addition, the number of signal counts per-emitter in the STORM experiment was significantly lower than in the telomere data, and followed a much less uniform distribution. Therefore, to take this into account we modelled the signal counts in STORM experiment using a Gamma distribution with a shape parameter of k=3 and a scale parameter of $\theta = 3000$ [counts]. Interestingly, for the STORM experiment, we found it beneficial to alter the GT labels and discard emitters with an extremely low number of counts (below 6K signal counts). This deliberately introduced "label-noise" allowed us to learn a more robust recovery net, coping easily with the non-uniform background introduced by dim/out of range emitters throughout the FOV.

Finally, in our implementation the training positions are randomly drawn within the 3D cube of possible locations. To improve the uniformity of volume coverage, we draw the continuous positions using two consecutive steps; First we discretize the volume to coarse voxels and randomly choose disjoint indices. Afterwards, each index is added to a random continuous shift within the voxel

and turned to microns using the coarse voxel-size. Note that all of this happens prior to network training. The boolean grid used as label in training is given by projecting the continuous positions on the recovery grid (voxel size of either $27.5 \times 27.5 \times 33\ nm^3$ or $27.5 \times 27.5 \times 50\ nm^3$). Although this strategy was simple and convenient in this work, a smarter training set generation can improve learning. For example, a biased sampling scheme with more probability to draw positions from the edges of the axial range (see Supplementary Video 4) can accelerate convergence, and potentially alter the learned mask, although care must be taken to not introduce artifacts.

4.2 Loss function

In computer vision, approaches for inferring the numerical coordinates of key-points in an input image are crudely divided into two classes: approaches that try regressing the coordinates directly using fully-connected (FC) layers (e.g. [30, 47]), and approaches that project the coordinates to the grid using a soft representation (e.g. a heatmap [12, 48]), and afterwards employ representation-matching. The former suffer from two fundamental drawbacks:

1. FC layers limit the model applicability to specific spatial dimensions which necessitates additional manipulation to handle images of general dimensions.

2. FC layers lack inherent spatial generalization [49], which is the ability to generalize knowledge attained at one location during training to another at inference time. This is one of the reasons why augmentation techniques, such as horizontal and axial shifts, are useful for training classification models.

Moreover, a grid representation avoids the inefficient learning of a non-linear mapping from feature space to emitter positions, and provides meaningful voxel-wise supervision. Hence, while FC layers can potentially provide more accurate coordinates, they do not have the generalization ability afforded by spatially shared parameters and are prone to over-fitting [50]. Therefore, we adapt a discrete representation approach, and project the continuous coordinates to the grid.

Next, two alternative approaches can be considered to tackle the task of localization using a CNN. Namely, one approach is to think of localization as a binary classification problem where the CNN predicts a binary occupancy volume, such that 0 denotes an empty/vacant voxel and 1 denotes an occupied voxel containing an emitter. A widely used loss function in this case is the cross-entropy (CE) loss. Although, even for dense localization, the vacant and occupied voxels are highly imbalanced, with only few voxels containing emitters. Therefore, the CE loss is usually either weighted [51], replaced with a Focal loss [52], or applied to a "blobbed" version of the desired boolean volume e.g. by placing a disk around each GT position [53–55]. Afterwards in post-processing, the CNN prediction is usually thresholded, and the final prediction is given by a centroid/local-maximum operation. Alternatively, a second approach is to consider a soft version of the binary classification problem and take a regression route. Namely, by placing a small Gaussian around each GT position (e.g. with std of 1 voxel), we can match continuous heatmaps via an ℓ_2 loss [11, 12]. Heatmap matching usually provides more meaningful gradients and ease the learning process convergence. Here, our loss function ℓ is a combination of two terms:

$$\ell\left(y,\hat{y}\right) = \left\|y \circledast g_{3D} - \hat{y} \circledast g_{3D}\right\|^2 + \lambda\left(1 - 2 \times \frac{\sum_{i=1}^{N} y_i \hat{y}_i}{\sum_{i=1}^{N} y_i + \sum_{i=1}^{N} \hat{y}_i}\right) \tag{S26}$$

Where $y, \hat{y}$ are the ground truth (GT) and the predicted boolean grid respectively, g_{3D} is a 3D Gaussian kernel with a standard deviation of 1 voxel, λ is a regularization parameter, and N is the number of voxels in the prediction grid.
The first term is a heatmap matching term where we measure the proximity of our prediction to the simulated GT by measuring the ℓ_2 distance between their respective heatmaps. As for the second term, it is a measure of overlap which provides a soft approximation of the true positive rate in the prediction. Note that this measure doesn't take into account false positives, and hence if optimized alone will result in a predicted volume of 1s. Although, with our loss function this is not a feasible solution as it is not favored by the first term. The two terms are weighted with a regularization parameter $\lambda = 1$ determined empirically. In addition, we weight voxels containing emitters with a factor of $W = 800$ in order to balance out the contributions of vacant and occupied voxels throughout training. Hence, the CNN output is constrained to be in the range $[0, 800]$. This strategy makes optimization easier and prevents gradient clipping.

Note that optimally the second term should be replaced with a soft approximation of the Jaccard loss [56] or dice loss [57, 58] which are the metrics we are ultimately interested in optimizing. However, although recent results on approximating the Jaccard loss look promising [59], the high class imbalance between empty and occupied voxels make the optimization process challenging.

To conclude, while the proposed loss function (equation (S26)) led to satisfactory results, more optimized choices could further improve performance, for example, a multi-scale approach such as [30]. Alternatively, a non static Gaussian kernel that shrinks over epochs could accelerate convergence. Finally, recent works [49, 60, 61] have suggested bridging the gap between coordinate regression and heatmap matching via the *soft-argmax* function. While in their current version these works assume a known fixed number of key-points and predict a volume per point which is not feasible for localization microscopy, future extensions might prove valuable.

4.3 Optimization and hyper-parameters

We used the Adam optimizer [62] with the following parameters: lr $= 5 \times 10^{-3}, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$. The batch size was 16 for learning a phase mask, and 4 for learning a recovery net (due to GPU memory). We experimented with Group Normalization (GN) [63] as an alternative to Batch Normalization (BN) [7] for the smaller batch size, but found that BN gave consistently better results. The learning rate was reduced by a factor of 10 when the loss plateaus for more than 5 epochs, and training was stopped if no improvement was observed for more than 10 epochs, or alternatively a maximum number of 50 epochs was reached. The initial weights were sampled from a uniform distribution on the interval $\left[-\sqrt{k}, \sqrt{k}\right]$ where $k = \frac{1}{k_x \times k_y \times C_{in}}$, with k_x, k_y the filter spatial dimensions, and C_{in} the number of input channels to the convolutional layer. No further regularization was used (e.g. weight decay [64] or dropout [65]). Training and evaluation were run on a standard workstation equipped with 32 GB of memory, an Intel(R) Core(TM) $i7-8700$, 3.20 GHz CPU, and a NVidia GeForce Titan Xp GPU with 12 GB of video memory. Phase mask learning took ≈ 25 h, and recovery net training took ≈ 35 h. Our code is implemented using the Pytorch framework [40], and is made publicly available at `https://github.com/EliasNehme/DeepSTORM3D`.

5 Post-processing

The final list of localizations is given by the 3D Center of Gravity (CoG) estimator applied to local maximas in the prediction volume that are above a chosen global threshold. While it is possible to use more sophisticated post-processing steps we choose to use this simple and efficient strategy to keep our method as fast as possible. This is extremely important for 3D STORM experiments covering large axial ranges, as these normally entail processing a few tens of thousands of frames. To implement our strategy on GPU, we use the following 4 steps for post-processing:

1. First, the CNN prediction volume is thresholded using the function `torch.where`, with a global threshold normally in the range $[40, 160]$. The appropriate choice of the threshold is dependent on the input image Signal-to-Noise Ratio (SNR) and on the accuracy of the PSF model. For example, if the input image has a relatively low SNR (e.g. $\approx 9K$ signal counts with ≈ 150 background counts), or alternatively the training set was generated using the theoretical phase mask rather than a retrieved pupil function, the optimal threshold is more likely to be 40.

2. Second, we discard predictions that are not local maxima in their 3D vicinity. The number of neighboring voxels in the 3D vicinity of the peak was chosen such that the 3D radius for peak finding was $r_{peak} = 100$ nm for the STORM experiment and $r_{peak} = 150$ nm for the telomere data. To run this step efficiently on GPU, we compare the thresholded prediction volume to the result of applying the function `torch.nn.MaxPool3d` with a stride of 1 and a kernel size of $2r_{peak}$ in all three axis. Usually, for a high SNR input image with relatively mild overlaps this step is not necessary. However, This step is crucial for low SNR highly overlapping images, as often the net tends to predict small 3D "blobs" (3×3 cube of values), with the maximum being often in the underlying emitter position. While this step potentially limits the achievable resolution at low SNR, keep in mind that overlaps in 2D normally translates to non-overlapping "blobs" in 3D. Hence, this is merely a limitation for standard imaging experiments using a 2D detector.

3. Third, around each found local maxima we calculate a 3D CoG estimator in x, y and z, using the `torch.nn.functional.conv3d` function. The network confidence in each of the neighboring voxels is used as its relative weight.

4. Finally, we compile a list of localizations by translating the output of the CoG estimator to μms according to the recovery voxel-size (which is either $(27.5 \times 27.5 \times 33)\ nm^3$ for mitochondria, $(27.5 \times 27.5 \times 50)\ nm^3$ for fixed telomeres, or $(40 \times 40 \times 50)\ nm^3$ for live telomeres).

Next, let us discuss the effect of both the threshold T in step 1, and the peak finding radius r_{peak} in step 2. The effect of the threshold is quite straighforward (Fig. SN5.1a). The higher the threshold, the higher the net confidence in its prediction. This means we will improve the localization precision on the expense of detecting less emitters. On the other hand, if the threshold is too low we will detect more false positives, which will ultimately result in a lower jaccard. The sweet spot between the two extremes was $T = 40$ for the mitochondria sample, and $T = 160$ for the telomere sample. The result for scanning the threshold in the mitochondria conditions was omitted for brevity as it was similar in both cases. Of course, the optimal threshold is dependent on the density as well, but since the latter is not known beforehand, we choose to keep our method as simple as possible and used a global threshold regardless of the density.

As for the peak finding radius, its role is to group nearby localizations and thereby lower the amount of false positives. A very big radius will throw away true positives, and therefore lower the jaccard index (Fig. SN5.1b,c left panel). On the other hand, a very small radius will result in a large amount of false positives, and thereby still lower the jaccard index. As for the localization precision, the size of the radius will affect the CoG estimator employed in step 3. This will have a different effect depending on the SNR (Fig. SN5.1b,c middle and right panels). At high SNR, a radius of $r_{peak} \in [2\Delta_i, 5\Delta_i]$ voxels improves the axial localization precision by $\approx$10 nm. On the other hand, for lower SNR this step decreases the axial localization precision by $\approx$8 nm as it might average nearby true positives axially. Although, we saw its necessary to prevent a high number of false positives in STORM experiments.

Finally, the results illustrated in Fig. SN5.1 are for networks trained with the Tetrapod PSF, although the effects are similar for other PSFs, and therefore omitted for brevity.

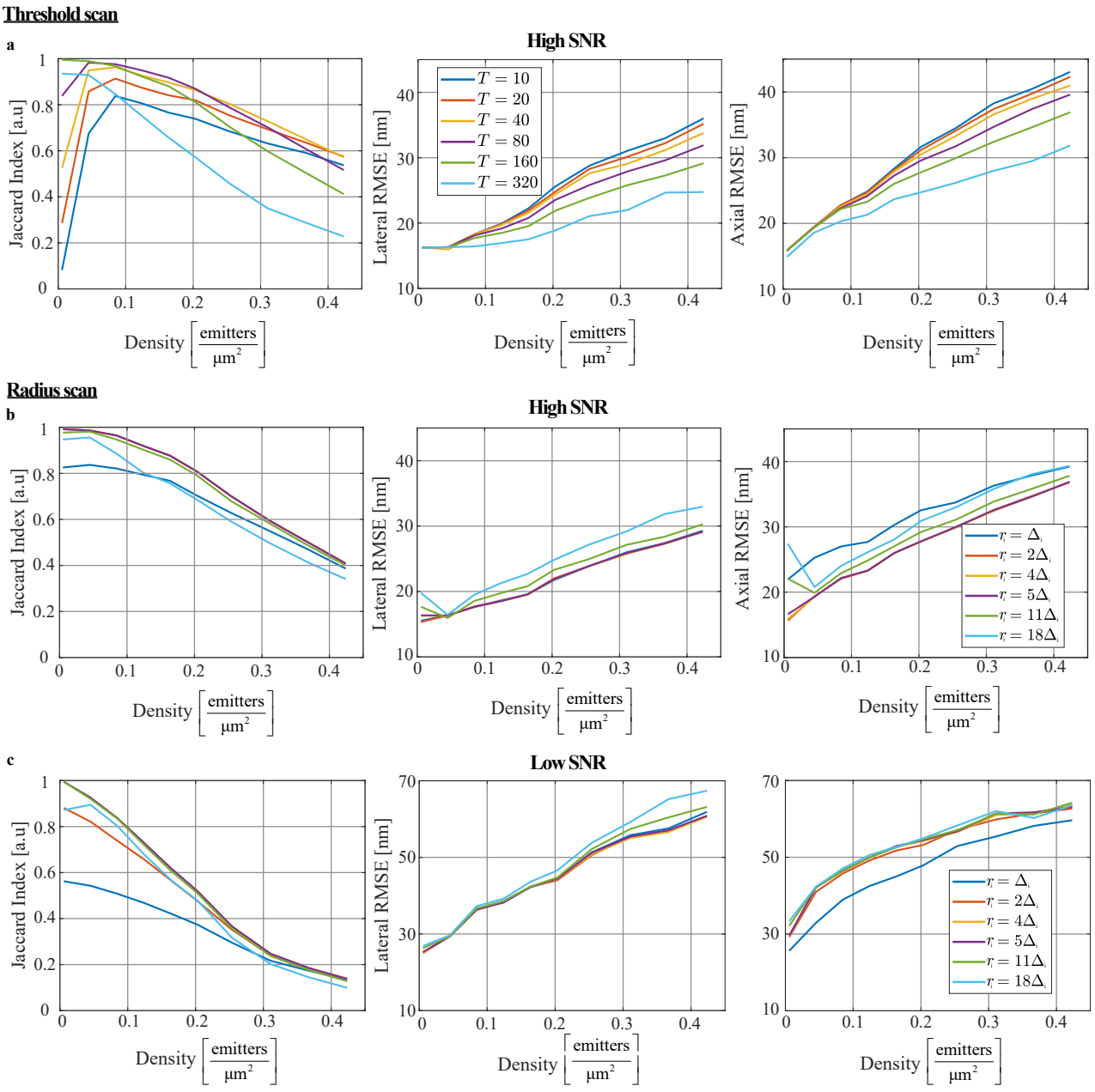


Fig. SN5.1. Post-processing parameters. **a** Jaccard index and lateral \axial RMSE as function of the global threshold T at high SNR. The peak finding radius was fixed to 4 voxels, and the SNR conditions were similar to the telomere experiment.**b** Jaccard index and lateral \axial RMSE as function of the peak finding radius r_{peak} at high SNR. The radius is reported in voxels, and is translated to nm according to the respective voxel side ($\Delta_{xy} = 27.5$ nm, $\Delta_z = 50$ nm). The threshold was fixed to T=160, and the SNR conditions were similar to the telomere experiment. **c** Jaccard index and lateral \axial RMSE as function of the peak finding radius r_{peak} at low SNR. The radius is reported in voxels, and is translated to nm according to the respective voxel side ($\Delta_{xy} = 27.5$ nm, $\Delta_z = 33$ nm). The threhsold was fixed to T=40, and the SNR conditions were similar to the STORM experiment.

6 Assesment metrics

To compare localizations directly, we first need to solve the assignment problem [66], meaning, we need to match each recovered position $\left(x_i^{rec}, y_i^{rec}, z_i^{rec}\right)$ to a nearby ground truth (GT) position $\left(x_j^{gt}, y_j^{gt}, z_j^{gt}\right)$ such that the overall euclidean distance between matched points is minimized. The matching was computed using the Hungarian algorithm [66] with a threshold distance of 150 nm to rule out False Positives (FP). Recovered points that were matched to a GT point were regarded as True Positives (TP). And finally, GT points that were not matched were regarded as False Negatives (FN). Next, following [4] we computed three standard metrics to compare two sets of points:

a. Jaccard Index (JI) defined as:

$$\mathrm{JI} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{S27}$$

This metric measures the fraction of correctly identified points in a dataset. A Jaccard index of 1.0 means perfect detection without spurious FPs. It is particularely important for localization techniques to detect a large fraction of the molecules in each frame, as this ultimately dictates the amount of needed frames (e.g. for super-resolution imaging) or more extremely the amount of needed experiment repetitions (e.g. for extracting the diffusion coefficient from single particle tracking trajectories).

b. Root Mean Squared Error (RMSE) in both the lateral (xy) and the axial (z) dimensions defined as:

$$\text{Lateral RMSE} = \sqrt{\frac{1}{TP}\sum_{i \in S_{TP}} \left(x_{m(i)}^{rec} - x_i^{gt}\right)^2 + \left(y_{m(i)}^{rec} - y_i^{gt}\right)^2} \tag{S28}$$

$$\text{Axial RMSE} = \sqrt{\frac{1}{TP}\sum_{i \in S_{TP}} \left(z_{m(i)}^{rec} - z_i^{gt}\right)^2} \tag{S29}$$

Where $m\left(i\right)$ is the index of the recovery point matched to GT point i, and S_{TP} is the set of matched GT points. These two metrics quantify the precision the localization algorithm, and ultimately determine the achievable resolution. In contrast to the Jaccard index, the RMSE is computed only for TPs and lower is better. Moreover, the lowest achievable precision for an unbiased localization algorithm is bounded by the Cramer-Rao Lower Bound [67].

Although it is possible to define a metric unifying equations (S27), (S28), and (S29) to a single number including also the software runtime [4], throughout this paper we report all 3 metrics separately for convenience.

7 Modified matching pursuit

The approach presented below was first described in the supplementary information of [3], and is closely related to [31, 68–70]. Before we go into details, let us first describe the Maximum Likelihood Estimator (MLE) for fitting single emitters which this method builds upon.

7.1 Maximum likelihood estimation

MLE is a technique for estimating the parameters of a statistical model based on a set of experimental observations, assuming we know the underlying noise model [67]. Specifically, given the imaging model PSF I (equation (S13)), the Poisson noise model (equation (S14)), a measured PSF of a single emitter y, assuming i.i.d. pixel measurements the likelihood function $\mathcal{L}$ is given by [71]:

$$\mathcal{L}\left(\Theta; y\right) = \prod_{i=1}^{M} \frac{I_i\left(\Theta\right)^{y_i} e^{-I_i}}{y_i!} \tag{S30}$$

Where $\Theta = \left(x_0, y_0, z_0, N, b\right)$ is the unknown emitter 3D position and local SNR, and M is the number of measured pixels. Therefore, the ML estimator is given by:

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmin}}\left(-log\left(\mathcal{L}\left(\Theta; y\right)\right)\right) \tag{S31}$$

$$= \underset{\Theta}{\operatorname{argmin}} \sum_{i=1}^{M} I_i - y_i \ln\left(I_i\right) \tag{S32}$$

Where the likelihood maximization problem is exchanged with the equivalent negative log-likelihood minimization problem, and the latter is solved conveniently via MATLAB's fmincon routine. This approach is known to achieve results close to the theoretical limit also known as Cramer-Rao Lower Bound (CRLB) [67], and is considered the gold standard for single emitter fitting [71], with available efficient implementations utilizing GPU acceleration [27, 33, 42]. However, for multi-emitter fitting, and specifically for the case of z-dependent PSFs, this approach becomes computationally prohibitive, and alternative approaches need to be explored. An example family of well-performing methods [31, 68–70] are approaches based on "sequential-fitting" as described next.

7.2 Continuous matching pursuit

Matching Pursuit (MP) is a method that relies on a sequential fit-and-subtract routine commonly used for sparse signal recovery [72]. Usually, MP is discussed in a discrete setting with a fixed number of possible "atoms" (e.g. PSFs) that can be combined to comprise the measured field-of-view (FOV). Here, we apply a continuous variant of MP, enabled because our dictionary is given by a continuous generative model of the PSF (equation (S13)). The basic idea is to decouple the multi-emitter fitting problem into sequential single-emitter fitting sub-problems, where in each iteration we fit the emitter that is most correlated with the residual. Next, the fit result is subtracted and the residual is updated. This process is iterated till a convergence criterion is met.

More formally, first, we start by creating a coarse dictionary $\mathcal{D}$ with atoms a_k comprised of the model PSF I_r (equation (S13)) sampled at $r_k = (x_0 = 0, y_0 = 0, z_0 = k\Delta_z)$, with $\Delta_z = 200$ nm, $k \in \{0, ..., 20\}$. The atoms are normalized to have a unit ℓ_2 norm: $a_k^{norm} = \frac{a_k}{\|a_k\|}$. Second, we set the residual R to be the measured image y normalized to have a unit ℓ_2 norm: $R = \frac{y}{\|y\|}$. Next, we initialize the set of recovered locations S, and in each iteration we repeat the following steps:

1. Calculate the normalized correlation volume with the residual defined as:

$$N_{corr}[m, n, k] = R[m, n] \circledast a_k^{norm}[-m, -n] \quad \forall k \in \{0, ..., 20\} \tag{S33}$$

2. Find the maximally correlated PSF from the dictionary in the coarse 3D grid:

$$\left(\hat{m}, \hat{n}, \hat{k}\right) = \underset{m,n,k}{\operatorname{argmax}} N_{corr}[m, n, k] \tag{S34}$$

3. Crop a fixed region from the residual R around the coarse localization from the previous step:

$$R_c = R\left[\hat{m} - \Delta_{xy} : \hat{m} + \Delta_{xy}, \hat{n} - \Delta_{xy} : \hat{n} + \Delta_{xy}\right] \quad \text{with} \quad \Delta_{xy} = 25 \text{ [px]} \tag{S35}$$

4. Fit the cropped residual R_c using MLE (equation (S32)) initialized with $\left(\hat{m}, \hat{n}, \hat{k}\right)$ to refine the emitter 3D position and estimate the signal and background counts:

$$\hat{\Theta} = \left(\hat{x}_0, \hat{y}_0, \hat{z}_0, \hat{N}, \hat{b}\right) = \underset{\Theta}{\operatorname{argmin}} \left(-log\left(\mathcal{L}\left(\Theta; R_c\right)\right)\right) \tag{S36}$$

5. Update the set of recovered emitter positions:

$$S = S \cup (\hat{x}_0, \hat{y}_0, \hat{z}_0) \tag{S37}$$

6. Calculate the emitter model image I_{emitter} using equation (S13) with the estimated parameters $\hat{\Theta}$:

$$I_{\text{emitter}} = \hat{N} \times \frac{I_{\hat{r}=(\hat{x}_0, \hat{y}_0, \hat{z}_0)}[m, n]}{\sum_m \sum_n I_{\hat{r}}[m, n]} + \hat{b} \tag{S38}$$

7. Subtract I_{emitter} to update the residual for further fitting:

$$R = R - I_{\text{emitter}} \tag{S39}$$

Convergence is achieved when either the mean residual drops below a threshold (e.g. mean background per-pixel), or the overall estimate I_S correlation with the measured image plateaus.

Note first that the run-time and amount of computations grow linearly with the number of emitters in the field-of-view. Hence, the approach is extremely inefficient for dense fields of overlapping emitters. Second, the strategy taken in step 4 is sub-optimal since the images of overlapping emitters are not explained well by single-emitter fitting. One famous extension of MP is the Orthogonal Matching Pursuit (OMP) method [73] where in each iteration all accumulated emitters in the set S are re-fitted. In our case this approach is computationally prohibitive, and is not trivially implemented using MATLAB's fmincon. Finally, note that for a measured image with a single-emitter, the approach above reduces to single-emitter fitting with MLE, and hence is more accurate than our CNN which is limited by the resolution of the output grid (first data point in Fig. 2a main text). This is because our method was tailored to handle high emitter densities by bounding the precision for the single-emitter case. Although, as shown by recent works [44, 74], CNNs designed specifically for single-emitter fitting can achieve precision comparable to that of MLE. Hence, a cascaded approach combining our method with one of [44, 74] could be considered for further accuracy improvement.

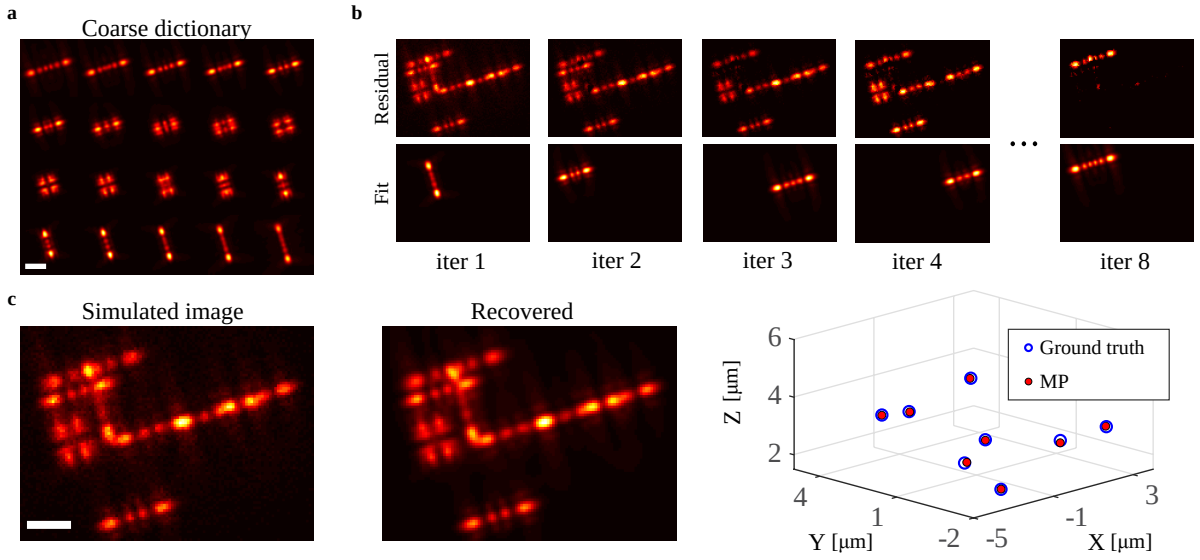


Fig. SN7.1. Continuous matching pursuit. **a** Coarse dictionary with 20 atoms a_k cenetered in xy and spaced with 200 nm steps in z. **b** Example unfolding of the MP iterations (residual - top, fitted emitter image - bottom) for a simulated image with 8 overlapping emitters. **c** Left: Input image is compared with the overall estimated image I_S by MP. Right: 3D comparison of the simualted emitter positions and the set of recovered positions by MP S. Scale bar is 3 μm.

7.3 Comparison at low SNR

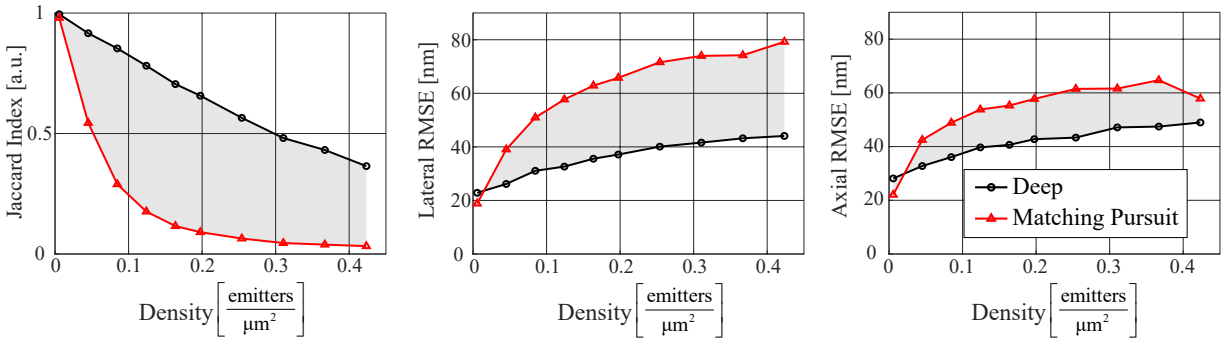


Fig. SN7.2. Comparison to MP at low SNR. **a** The trained CNN is superior to the matching pursuit approach in both detectability (Jaccard index) and in precision (Lateral\Axial RMSE). Matching of points was computed with a threshold distance of 150 nm using the Hungarian algorithm [66]. Each data point is an average of $n = 100$ simulated images. Average standard deviation in Jaccard index was $\approx 7\%$ for both methods, and average standard deviation in precision was ≈ 7 nm for the CNN, and ≈ 18 nm for MP. The SNR conditions were set similar to the STORM experiment, i.e. 9K signal counts, 20 counts per-pixel Poisson background, and a read noise with a standard deviation of 10 counts.

8 EPFL 3D challenge

8.1 DH high density modality

To put DeepSTORM3D into context with the plethora of existing high density methods, we applied it to the DH high density modality from the EPFL 3D challenge [4].

To train a localization net, first we needed to recover the phase mask representing the observed experimental DH PSF provided in the calibration bead-stack. To this end, we used the VIPR [46] phase retrieval method. Afterwards, the training examples were generated with a similar SNR and emitter density to the corresponding EPFL test set.

The entire axial range in this competition was ≈1.5 μm. To achieve maximal accuracy in z, we trained the models to output a grid of $D = 100$ channels in z, corresponding to an axial voxel-size of $\Delta_z = 15$ nm. The lateral voxel-size was $\Delta_{xy} = \frac{100nm}{4} = 25$ nm. The maximal dilation rate was $d_{max} = 4$ according to the DH lateral footprint, and the post-processing parameters were set to T=40 and $r_{peak} = 4$ voxels following a similar analysis as in section 5. In addition, prior to localization by the net, we subtracted the minimum value per pixel across the entire test stack to get rid of the non-uniform auto-fluorescence background component.

As for the training locations, we experimented with two different sampling schemes:

- Similarly to the rest of this work, we sampled the training locations uniformly at random in 3D. We used 9K images for training, and 1K images for validation.
- We used the training locations of the matching training set on the EPFL website. These locations were positioned along lines (simulated microtubules), and had a very similar clustering in space to the test set. To create a large enough training set, we augmented these positions by a factor of 4, using rotations in xy and flipping/scaling in z. The resulting image library was composed of 10K images with their corresponding emitter positions. These were then split to 9K images for training and 1K images for validation.

The test set was composed of 3125 frames of a $6.4 \times 6.4\ \mu m^2$ FOV, and was analyzed in ≈2 mins. In the random sampling case (Fig. SN8.1a) we managed to recover 14,168 emitters, whereas when we trained on emitters positioned along lines (Fig. SN8.1b) we were able to recover 18,347 emitters. Naturally, the reconstruction was more continuous (Fig. SN8.1b inset (i)), and crisper (Fig. SN8.1b insets (ii)-(iii)) when training on lines. However, this result needs to be considered with caution. While for the purpose of the competition this model will outperform the model trained on emitters positioned randomly in space, keep in mind that it was implicitly trained on the structure, and is therefore not expected to generalize well when tested on different structures. Nonetheless, it is worth mentioning that we did not optimize the sampling scheme for the training locations in this work, and the performance can be potentially boosted by sampling emitters that are randomly clustered in space.

Note that here we showed the results for the challenge data matching the SNR conditions encountered with Alexa647 (MT2N1HD). However, our submission also includes the results for SNR conditions matching mEos2/Dendra2 (MT4N2HD) and will be available online at the EPFL website: http://bigwww.epfl.ch/smlm/challenge2016/index.html?p=results, where it can be compared interactively with all other participants.

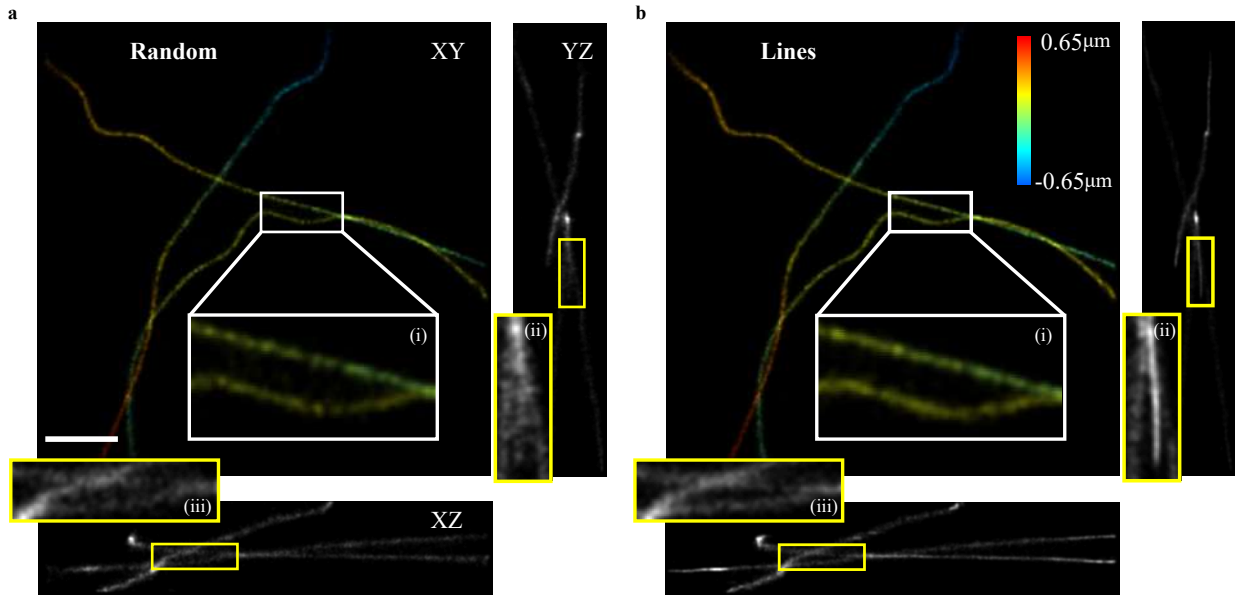


Fig. SN8.1. DH high density challenge. **a** Reconstructed test set when training with random emitters in space. **b** Reconstructed test set when training with emitters positioned along lines. In both cases, the 3D histogram is rendered with an isotropic 10 nm grid, and blurred with a gaussian with a standard deviation of 15 nm. The YZ and XZ projections are plotted in grayscale for convenience. Scale bar is 2 μm.

8.2 Comparison to SMAP-2018

We compared the Tetrapod-trained CNN to SMAP-2018 [42], which is a leading **single-emitter** fitting method that was also successful in localizing high-density of emitters [4].

To use SMAP-2018, we started by calibrating the spline coefficients in order to model the PSF. For this purpose, we simulated an axial stack of a bead PSF covering a 4 μm range with 10 nm steps. The calibration parameters used were the following: ROI size = 41 [px], distance between axial slices = 10 nm, no cross-correlation between slices to for alignment, filter size for peak finding = 8, relative cutoff = 1, smoothing factor = 1. Next, we used the calibrated spline model to localize emitters. For peak finding we used the maximal intensity projection PSF probed at the axial slice $z = 35$, with no additional smoothing ($s = 0$). The detection cutoff was set to the absolute (photons) mode with 24 photons, using the maximum criterion. Moreover, we used a rectangular ROI for fitting with 35 [px] sides. MLE fitting was done using the spline model coefficients with 60 iterations of the Levenberg–Marquardt algorithm per emitter. Moreover, for each emitter we initialized the fit with three different starting points in z ($z_0 = [-1, 0, 1]$ μm) and chose the solution with the maximum likelihood. Furthermore, we did not exclude the rim of the field-of-view (FOV) since some of the PSFs were touching the sides. To reject false positives and keep only precise localizations, we used the following filtering settings: xy-locprec = 100 nm, relative Log-likelihood = 2, iter $<$ $\max_{\text{iter}}$, and $\left| x_{fit} - x_{peakfind} \right| < 3$. Finally, one particularly useful filtering setting was the recovered photon number. We used a threshold of 24500 photons for the high SNR case (Fig. SN8.2a) and 4500 for the low SNR case (Fig. SN8.2b).

Note that in the low SNR case we could not get SMAP-2018 to recover all emitters even in the single emitter case (Fig. SN8.2b left panel). Although we tried different thresholds and peak finding settings (e.g. non-maximum suppression), no single setting was able to recover all emitters without introducing additional false positives and decreasing the jaccard index.

Nonetheless, SMAP-2018 is an excellent single-emitter fitting method. Here its performance was worse than MP since it was not designed to handle emitter overlaps, or PSFs with a varying lateral footprint in the axial dimension. Hence, SMAP being one of the leading software in dense 3D localization for other PSFs [4] highlights the importance of the method presented in this work.

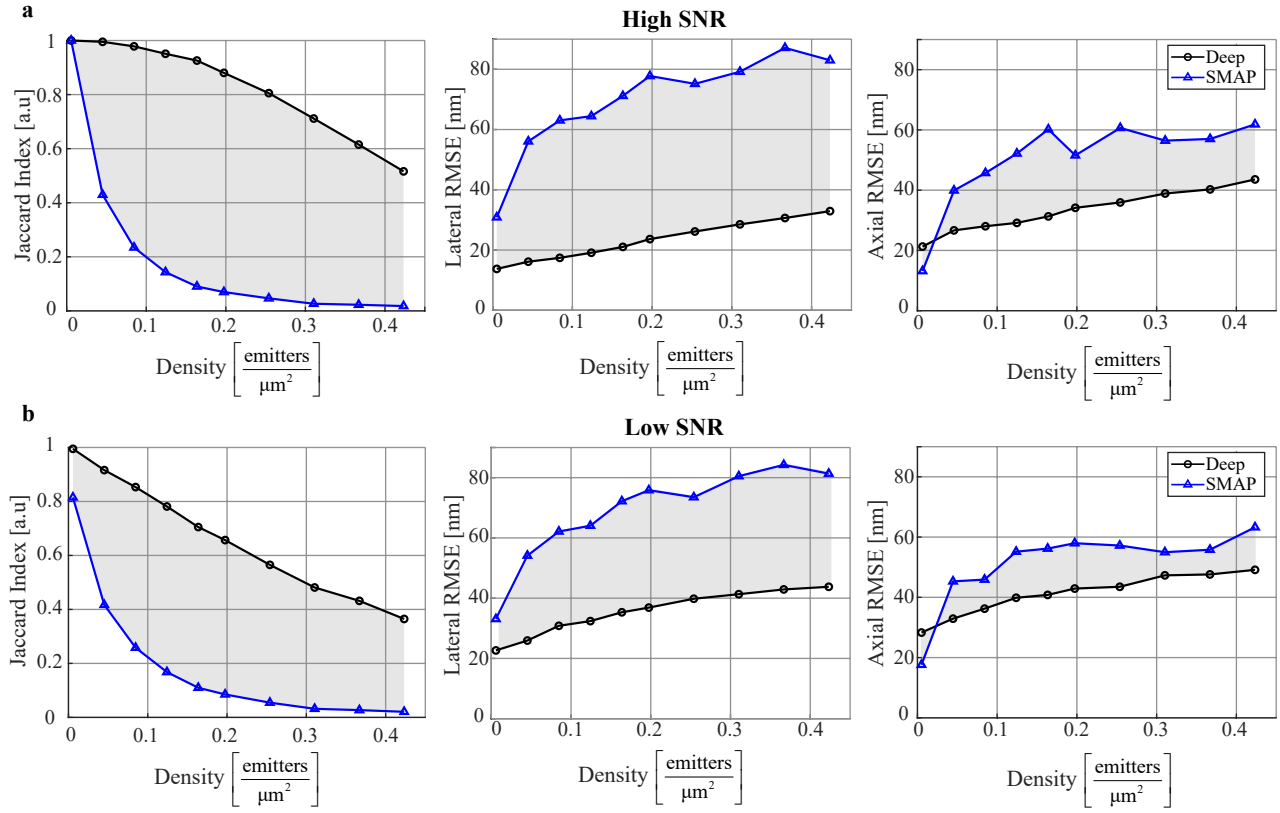


Fig. SN8.2. Comparison to SMAP-2018. **a** Jaccard index and RMSE comparison between a trained CNN (black) and SMAP-2018 (blue) at high SNR matching the telomere experiment. **b** Jaccard index and RMSE comparison between a trained CNN (black) and SMAP-2018 (blue) at low SNR conditions matching the STORM experiment. As expected, the trained CNN is superior to SMAP-2018 in both detectability (Jaccard index) and in accuracy (Lateral \Axial RMSE) at both high (**a**) and low (**b**) SNR. Matching of points was computed with a threshold distance of 150 nm using the Hungarian algorithm [66].

9 STORM imaging

9.1 Phase mask fabrication

For STORM imaging, the phase mask (PM) was fabricated in fused silica substrate through three iterations of photolithography, with Reactive Ion Etching(RIE) following each step. Chrome-Soda-lime masks were fabricated by a Direct Write Laser Lithography system (Heidelberg DWL66+). The fused silica substrate was coated with positive photoresist Az1518 and baked for 2 minutes at 90^0 C, with final thickness of 2.3 μm. The Karl Suss MA-6 was used as an exposure tool, with an exposure dose of 28 $\frac{mJ}{cm^2}$ UV light. Three hard mask patterns are prepared, one for each etching step. Next, the wafer was developed in TMAH: DI solution (concentration of 2.25%) for 55 seconds, then rinsed with DI water. After achieving the desired resist pattern, the fussed silica wafer was etched by CHF_3 plasma using a Plasma-Therm 790 RIE. Three steps of photolithography and etching to 140 nm, 280 nm and 560 nm resulted in 8 different heights from 0 to 980 nm, in steps of 140 nm.

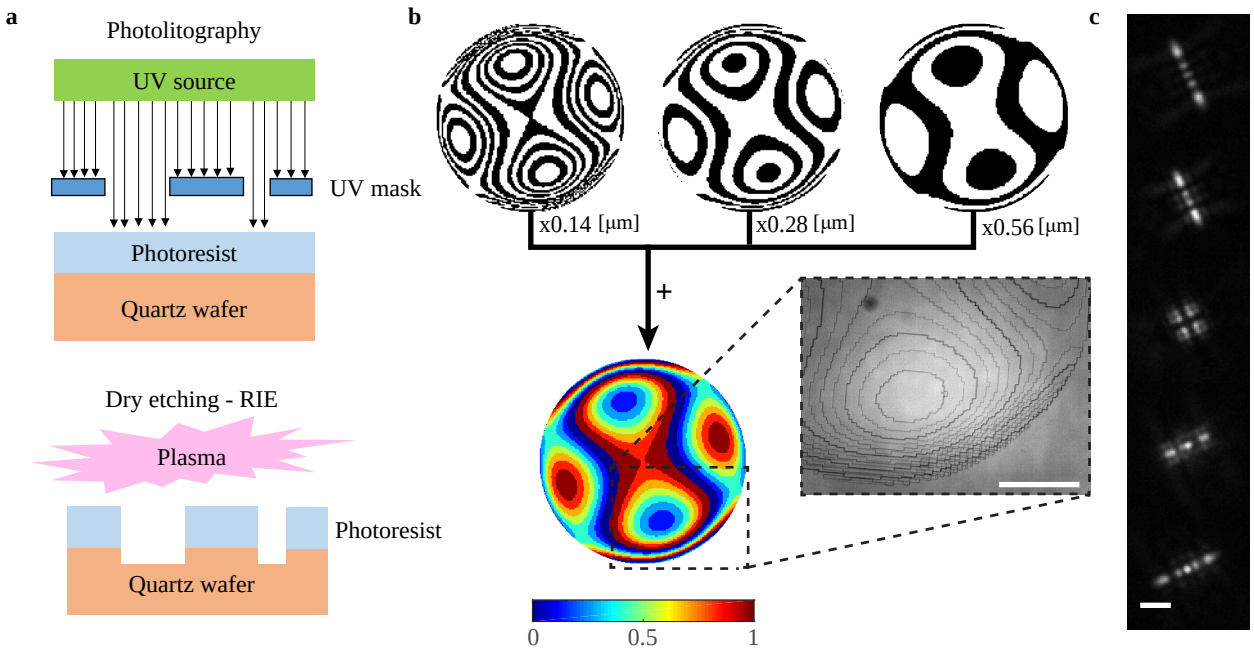


Fig. SN9.1. Phase mask fabrication. **a** In the photolitography step (top), a wafer coated with photoresist is illuminated through a hard UV mask. Afterwards, in the dry etching step (bottom), the wafer is etched according to the photoresist pattern. **b** The three UV masks used to generate the 3 corresponding height maps: 140 nm, 280 nm, 560 nm. Since the masks are stacked, the final mask includes 8 different heights, with steps of 140 nm (top). Zoom-in is an experimental measurement of the physical mask using a standard microscope (middle). Scale bar is 0.5 mm. **c** Measured z-stack of a bead on the coverslip with the physical mask. Scale bar is 2μm.

9.2 Density limit with Tetrapods

In order to test the performance limits of DeepSTORM3D with respect to the experimental density, we digitally summed consequential frames from the mitochondria experiment (Fig. 3 main text and Supplementary Videos 1-3). To create a data 2 times denser we summed frames 1 and 2, and then frames 3 and 4 etc. Similarly we summed each 4 and 8 consequential frames with no-overlaps to generate a data 4 and 8 times denser (Fig. SN9.2). The resulting number of frames in each dataset was 10K, 5K, and 2.5K respectively, corresponding to reconstruction times of ≈1h 40 mins, ≈45 mins, and ≈23 mins. The recovered number of localizations was ≈360K, ≈220K, and ≈90K respectively.

Note that DeepsTORM3D was able to recover the same number of localizations using frames sum (only ≈200 fewer emitters). Moreover, the summation actually improved the reconstruction at the right lower edge of the FOV (Fig. SN9.3a), where the counts of the same emitters across two consequential frames were summed to increase the effective SNR. In addition, note that this result was achieved within ≈1h 40 mins which is actually faster than the corresponding single-emitter method in ZOLA [33] that utilizes distributed MLE fitting on GPU.

As for the higher densities, the resolution seems to start deteriorating in the x4 case (Fig. SN9.3b), although relatively gracefully. While further extensions of DeepSTORM3D that explicitly account for the temporal dimension might bridge the gap to x4 denser samples, the x8 density (Fig. SN9.3c) seems much harder to achieve using the Tetrapod PSF.

In terms of acquisition time, while our exposure time in the actual experiment was 30 ms per-frame, in principle, the laser power can be amped up to blink the desired amount of emitters within 30 ms. Hence, this means the artificially constructed denser datasets (Fig. SN9.2) can be acquired within 5 mins, 2.5 mins, and 1.25 mins.

To put things in perspective with the current stat-of-the-art, methods achieving similar results relying on single-emitter Tetrapod fitting [33] require at least an $\approx \times 5$ longer acquisition time. Similarly, methods that combine the astigmatism PSF with axial scanning (e.g. [74]) would require ≈180K frames to cover a 4 μm axial range, resulting in $\approx \times 4$ longer acquisition even with a 14 ms exposure time. Finally, not only these numbers are conservative in favor of the methods mentioned above, but they are also calculated compared to analyzing the raw experimental frames, prior to summation. If we further compare them to the $\times 2$ denser dataset (Fig. SN9.3a), we get a speedup factor of $\times 10$ and $\times 8$ respectively, manifesting the unprecedented acceleration afforded by DeepSTORM3D.

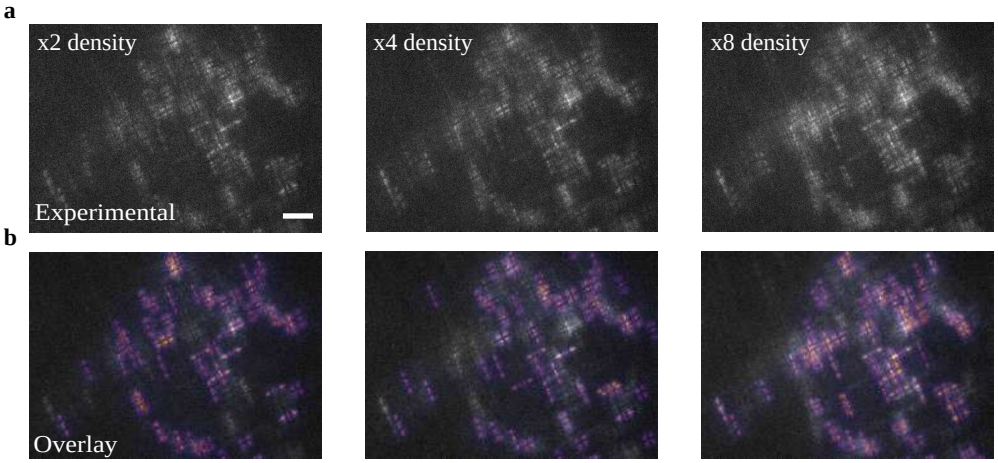


Fig. SN9.2. Increased experimental density. **a** Resulting frames from summing 2 (left), 4 (middle), and 8 (right) consequential experimental frames. **b** rendered frames from the corresponding 3D recovered positions by the CNN overlaid on top. Scale bar is 5 μm.

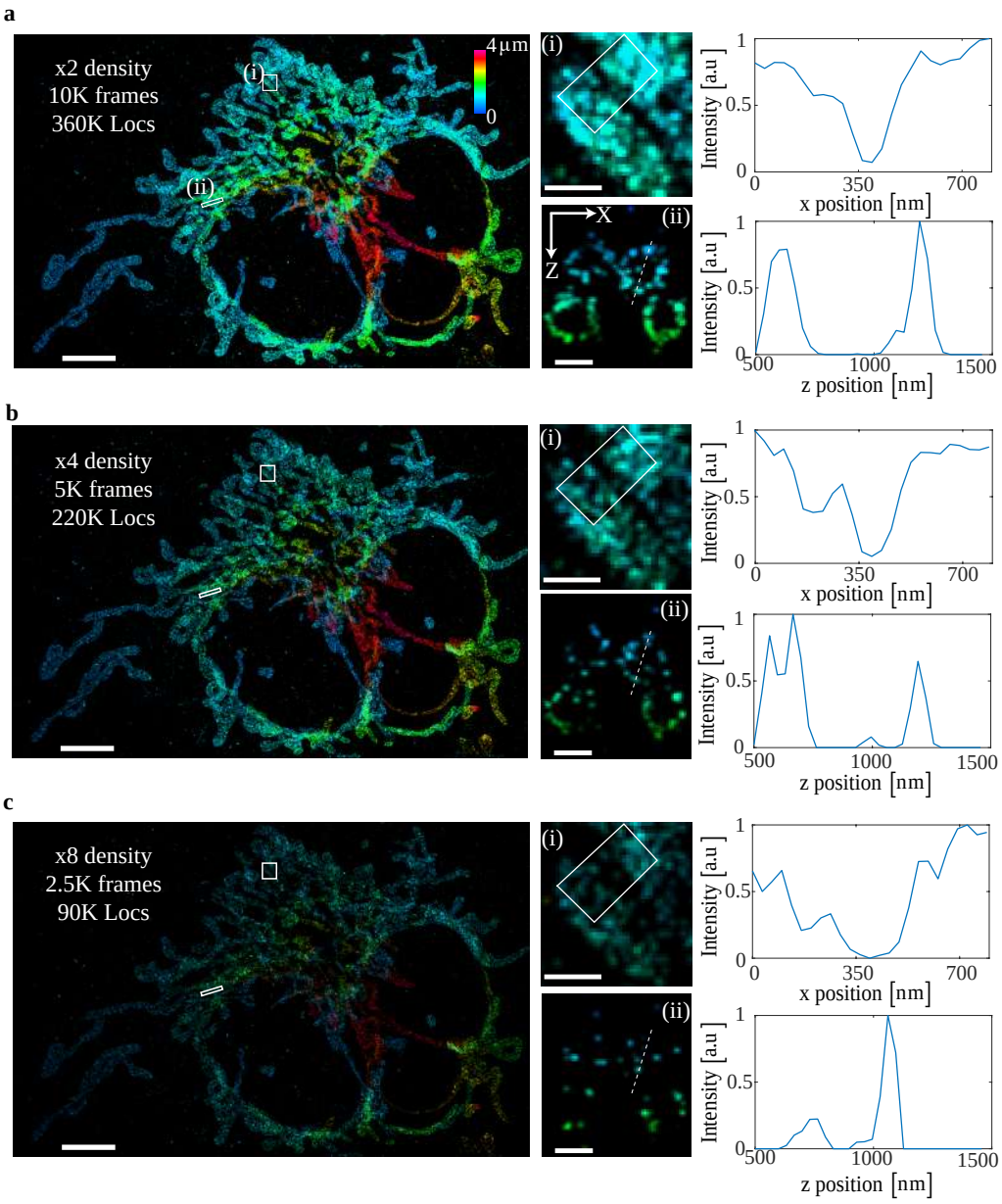


Fig. SN9.3. Super-resolution reconstructions as function of density. **a** Reconstruction from 10K frames ($\times 2$ density). **b** Reconstruction from 5K frames ($\times 4$ density). **c** Reconstruction from 2.5K frames ($\times 8$ density). Scale bar in the reconstruction is 5 μm, and scale bars in insets (i) and (ii) are 0.5 μm.

9.3 Resolution analysis

To estimate the resolution of our reconstructed super-resolved image, we simulated images with similar SNR using the retreived phase mask (Fig. SN9.4). The density of the emitters was varied in the range $\left[\frac{1}{\text{FOV}}, \frac{35}{\text{FOV}}\right]$ $\left[\frac{\text{emitters}}{\mu m^2}\right]$ with a field-of-view (FOV) of $13 \times 13\ \mu m^2$. To estimate the experimental density we used the number of localizations recovered by the CNN with a low threshold of $T = 10$. The resulting density was ≈0.1 $\left[\frac{\text{emitters}}{\mu m^2}\right]$ which means the expected resolution is ≈37 nm in xy and ≈50 nm in z.

The lateral resolution was also estimated directly from the mitochondria reconstruction using a recently published parameter-free method [75], and was found to be in great agreement with our simulation up to a single nanometer (Fig. SN9.4d), validating the accuracy of our resolution estimation.

To compare the result to the single emitter case we calculated the CRLB for a mean signal of 9000 $\left[\frac{counts}{emitter}\right]$, and a mean background of 140 $\left[\frac{counts}{pixel}\right]$. The result suggests that we achieve a factor of ≈2 relative to the CRLB in precision due to the combination of high density with a low SNR.

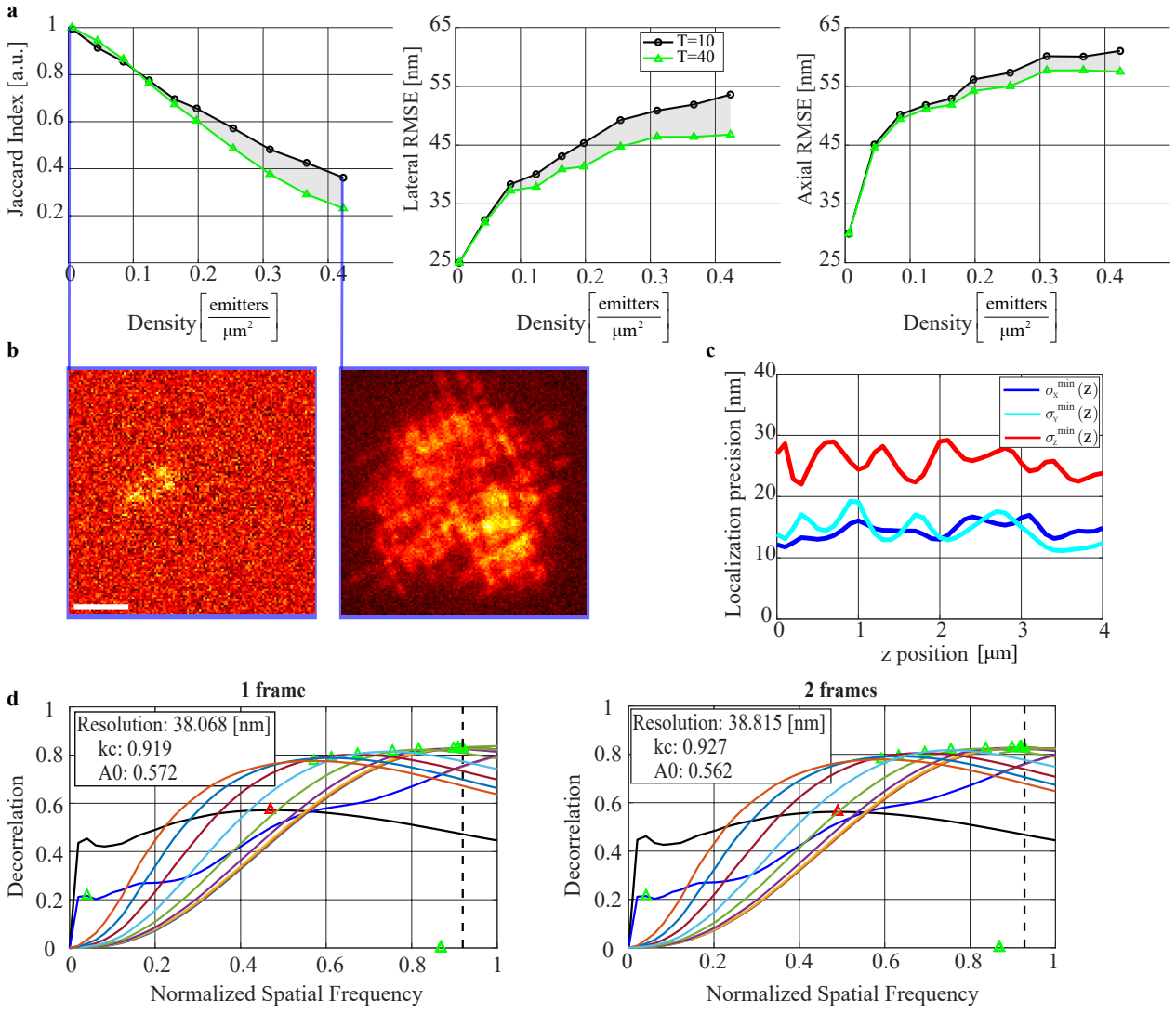


Fig. SN9.4. STORM resolution analysis. **a** Jaccard index and Lateral\Axial RMSE calculated at low SNR as a function of density, for two different thresholds $T = 10\backslash 40$. **b** Example frames with a single emitter (left) and 75 emitters (right). **c** CRLB calculated assuming $9K$ signal counts with 140 counts per-pixel background. Similarly to [25] differentiation is done numerically with 1 nm perturbations. **d** Decorrelation analysis of the mitochondria reconstruction from the original experimental frames (left) and from summing each 2 consequential frames (right). The estimated lateral resolution was ≈ 38/39 nm respectively. Scale bar is 3 μm.

10 Learned PSF analysis

10.1 Comparison to popular PSFs

The SNR conditions at which we learned the PSF were similar to the telomere experiment. Therefore, to put our learned PSF into context with respect to existing PSFs, we compared its performance against CNNs trained to localize the standard PSF, the DH PSF, the DH PSF after optimization using our method to extend its range to 4 μm (Fig. SN2.5), and the Tetrapod PSF. In accordance with each PSF lateral footprint, the maximal dilation rate d_{max} was set to 4, 4, 4, and 16 respectively. The comparison included a density scan with a fixed SNR (Fig. SN10.1a), and a SNR scan with a fixed density of 0.124 $\left[\frac{emitters}{\mu m^2}\right]$ (Fig. SN10.1b).

The results suggest that in the telomere SNR conditions our learned PSF (orange) outperforms all other PSFs, especially at high density (Fig. SN10.1a). Moreover, this conclusion in maintained across the entire range of signal counts ($[10K, 60K]$ this PSF was designed for (Fig. SN10.1b), however for lower signal counts (first 2 data points in Fig. SN10.1b), the axial localization precision is ≈10 nm worse than the Tetrapod and the DH PSF. For the optimization of this PSF to lower conditions see section 10.5.

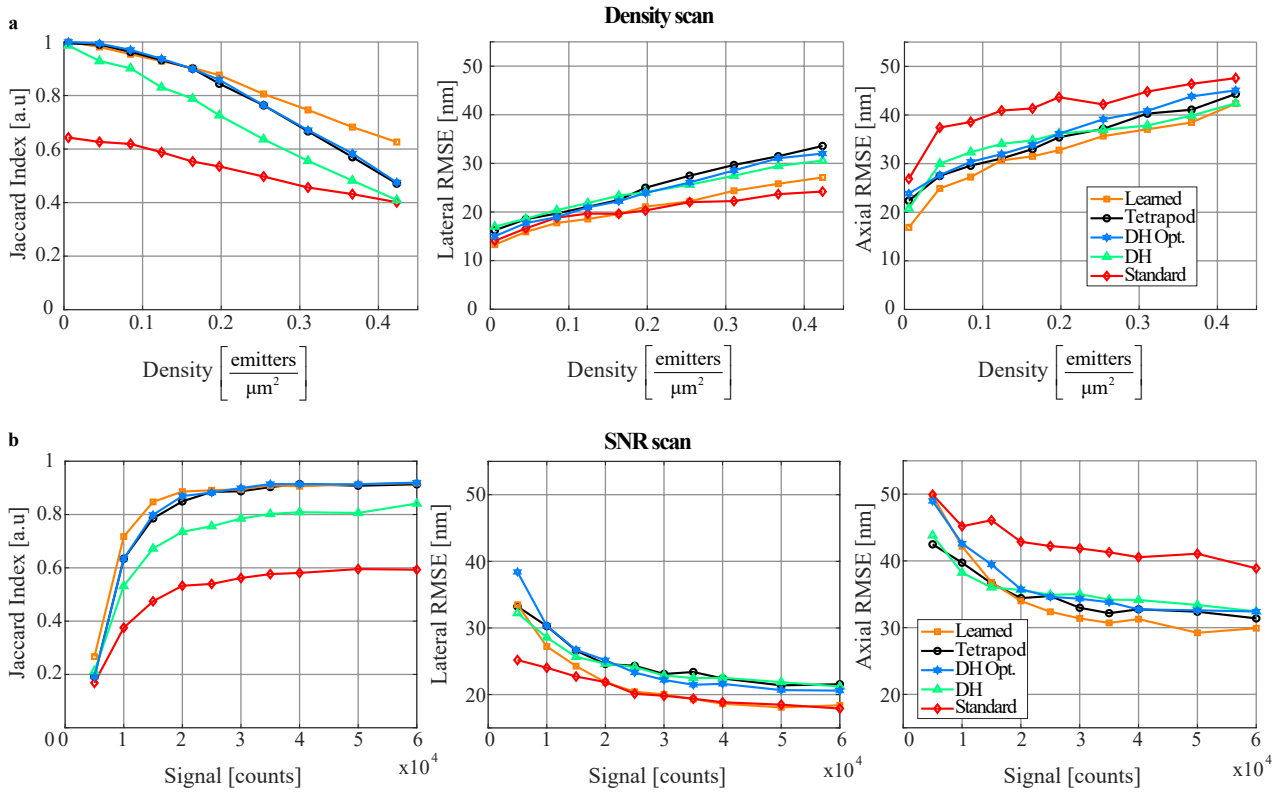


Fig. SN10.1. Comparison to other PSFs. **a** Comparison of the Jaccard index and the lateral \axial RMSE as function of emitter density between the Standard-PSF CNN (red), DH-PSF CNN (green), Leanred 4μm DH from Fig. SN2.5 (blue), Tetrapod-PSF CNN (black) and the Learned-PSF CNN (orange). The SNR conditions were similar to the telomere experiment. **b** Comparison of the Jaccard index and the lateral \axial RMSE as function of signal counts between the Standard-PSF CNN (red), DH-PSF CNN (green), Leanred 4μm DH from Fig. SN2.5 (blue), Tetrapod-PSF CNN (black) and the Learned-PSF CNN (orange) for a fixed density of 0.124 $\left[\frac{emitters}{\mu m^2}\right]$. Scale bar is 3 μm.

10.2 Implementation ease

Our system is composed of two main components: a 4f optical system, and a phase mask. Extending the image plane using a 4f optical system (Fig. 1a main text and Fig. SN3.1) is a relatively simple task, and has been explored thoroughly over the last decade [25, 27, 33, 76, 77]. As for the phase mask implementation, at first glance, our learned phase mask may seem challenging to implement as it contains a lot of phase jumps that are potentially challenging to achieve using commercial Spatial Light Modulators (SLMs). However, if we unwrap the phase (Fig. SN10.2) we get a discrete spiral with ≈3 main different values. Hence, implementing our PSF is straightforward with commercially available LC-SLMs.

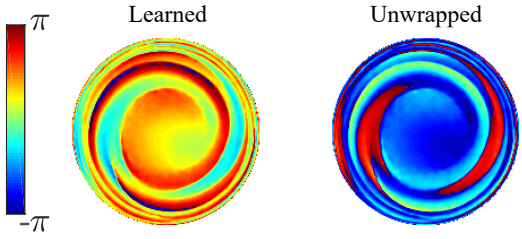


Fig. SN10.2. Learned mask unwrappig. The learned mask (left) can be unwrapped to a spiral with only ≈3 discrete values (right).

10.3 Sensitivity to lateral overlap

The learned PSF had a smaller lateral footprint compared to the Tetrapod PSF. This trait is extremely useful at high densities as it minimizes the probability of lateral overlap. However, a natural question in this case is whether this PSF is more vulnerable to lateral overlap, as smaller features are easier to confuse. To test this, we simulated emitters that are positioned nearby laterally with a growing axial separation (Fig. SN10.3). The lateral distance between the emitters was fixed to either $\Delta_{xy} = 0$ μm or $\Delta_{xy} = 0.5$ μm (diagonally).

When the two PSFs are at the same xy position, the learned PSF was harder to decode than the Tetrapod PSF especially at larger axial separations (Fig. SN10.3 jaccard index last data point). However, when the emitters were positioned $\Delta_{xy} = 0.5$ μm apart in the lateral dimension, the learned PSF quickly recovered and caught up with the Tetrapod PSF.

Finally, note that in our optimization the emitters were randomly sampled in space. Therefore, the case of zero lateral separation will happen with zero probability, which explains why the net will did not account for these negligible cases when designing the PSF. While potentially an additional loss function that ensures minimal correlation of the PSF across the axial dimension could further optimize such cases, in reality, such cases are difficult to localize with satisfying precision anyway, and are therefore negligible.

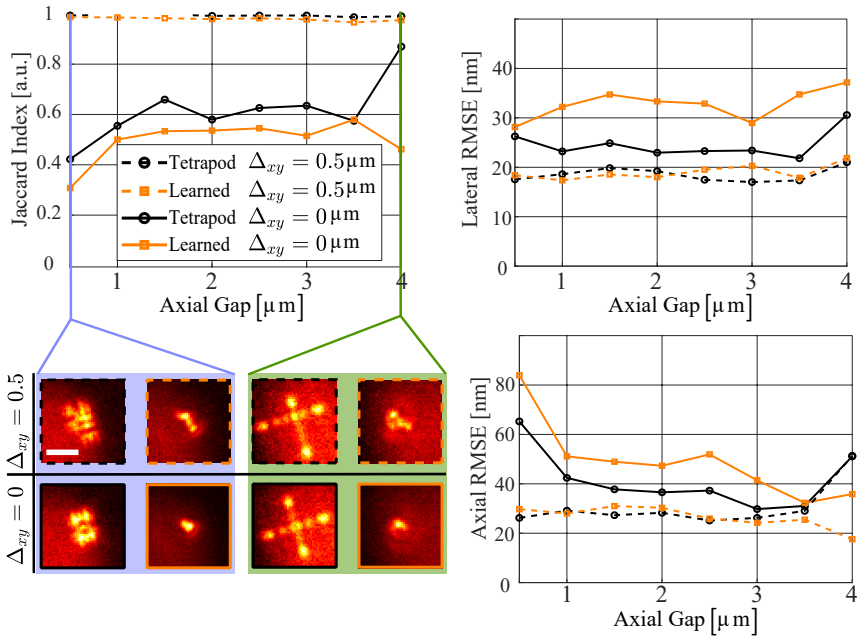


Fig. SN10.3. Lateral overlap. Comparison of the Jaccard index and the lateral\axial RMSE as function of axial gap between 2 emitters with the Tetrapod (black) and the Learned (orange) PSFs. Laterally, the emitters were positioned either in the same xy coordinates (continuous lines), or at a distance of 500 nm (dashed lines). Example PSFs at an axial gap of 0.5 μm and 4 μm are compared visually in both cases. Scale bar is 2 μm.

10.4 Experimental precision calibration

In cellular imaging (see section 12), precise ground truth is difficult-to-impossible to obtain. Therefore, to experimentally compare the precision of the learned PSF to the Tetrapod PSF, we scanned a fluorescent microsphere using 100 nm steps in the axial dimension. Since moving the microscope objective in oil is not interchangeable with moving the emitters in water, we acquired a larger axial range of ≈ 5 μm. In order to match the SNR of the measurement for both PSFs, we switched between the two masks in each axial step. Afterwards, the SNR of these measurements was digitally degraded to match the desired conditions. For the telomere conditions, the sum of the PSF was normalized to 30K counts, and afterwards a uniform background of 20 counts per pixel was added. The result was then passed through a per-pixel Poisson distribution and added to a read noise with a standard deviation of σ=10 counts. This was repeated 100 times at each axial position, each time with a different noise realization. For the STORM conditions the noise was kept the same, only this time the sum of the PSF was normalized to 9K counts.

In this experiment, it is of paramount importance to have an accurate PSF model as the precision of the localization net is directly related to the accuracy of the PSF model. Therefore, in parallel to this work, we have developed VIPR [46], a per-pixel phase retrieval method that is able to accurately model the learned phase mask. Here, we used VIPR to retreive both the Tetrapod and the learned masks in order to generate training examples for the corresponding CNN. Note that this is different from the method presented later in section 11 which we used only for the Tetrapod PSF in the telomere imaging experiments (Fig. 5 main text, and Fig. SN13.1,SN13.2).

For the telomere conditions (Fig. SN10.4), the results were in agreement with our simulations (Fig. 4 main text). Both PSFs had a similar performance, although they did not reach the CRLB due to our finite voxel-size (Fig. SN10.4c). However, for the STORM conditions (Fig. SN10.5), while the Tetrapod PSF reached the CRLB, the learned PSF was not able to reach the CRLB in the axial dimension. While we expected it to perform slightly worse than the Tetrapod for a lower SNR, in the single-emitter case (Fig. SN10.1b right panel), the loss in performance was worse than expected.

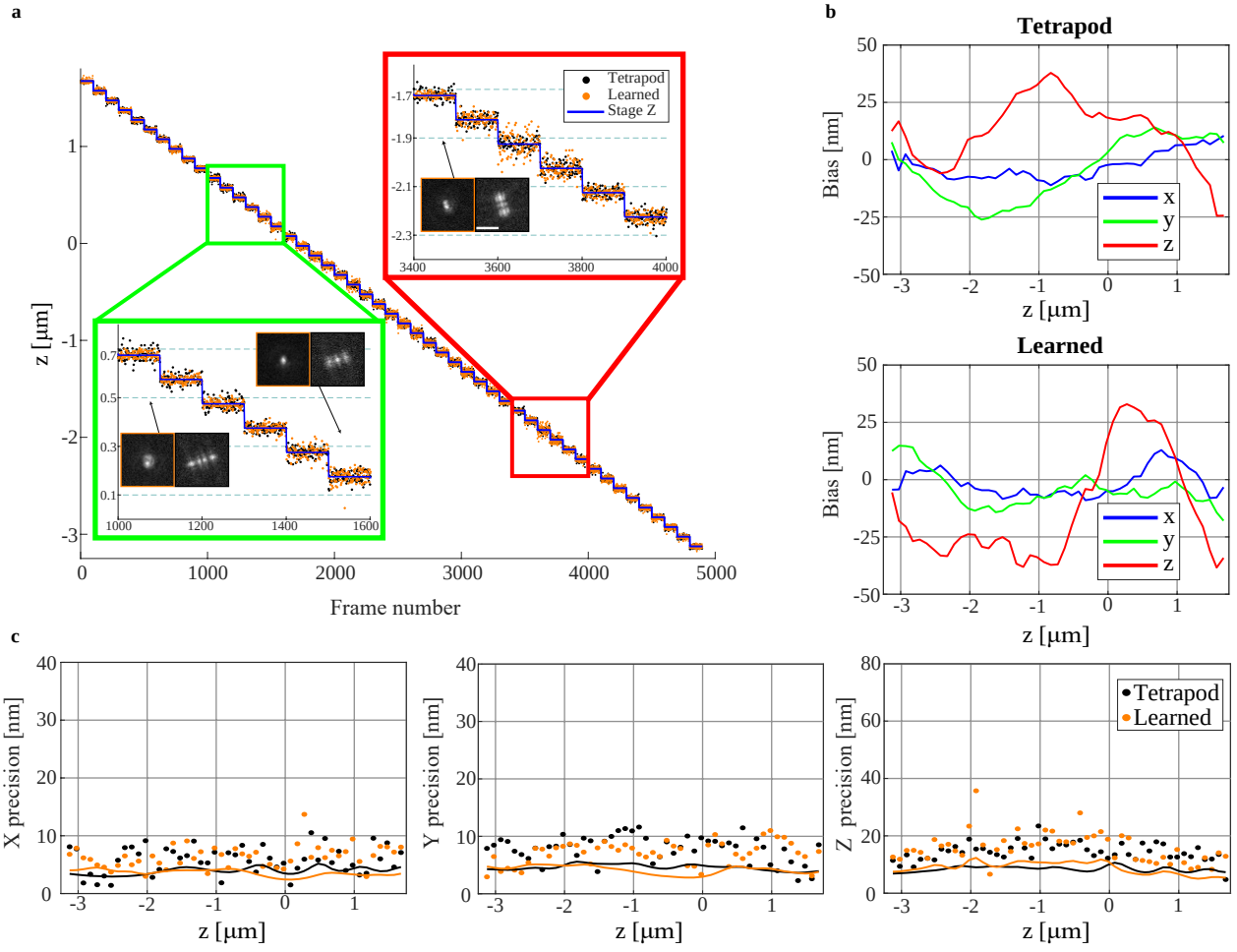


Fig. SN10.4. Experimental precision at high SNR. **a** CNN localizations for both the tetrapod (black dots) and the learned (orange dots) PSFs overlaid on top of the stage readout position (blue steps). each z position was localized with 100 different noise realizations. Example PSFs are shown at both the lower (green inset) and the higher (red inset) parts of the axial range. **b** calibrated bias according to the stage readout for both the Tetrapod (top) and the Learned (bottom) PSFs. **c** Calibrated precision in all three axis for both PSFs. dots mark the estimated precision and continuous lines mark the calculated CRLB. Scale bar in red inset in **a** is 2 μm.

This is due to the super-critical angle fluorescence (SAF) in the measurement on the coverslip (see Supplementary Video 6). This component had little effect in the high SNR case, as the network was able to capitalize on the subtle differences in the PSF. However, in the low SNR case these differences were below the detection limit causing the precision to drop. Nonetheless, when imaging in cells (> 1 μm) the SAF light effect vanishes, hence, the axial localization precision for the learned PSF is expected to improve and reach the CRLB.

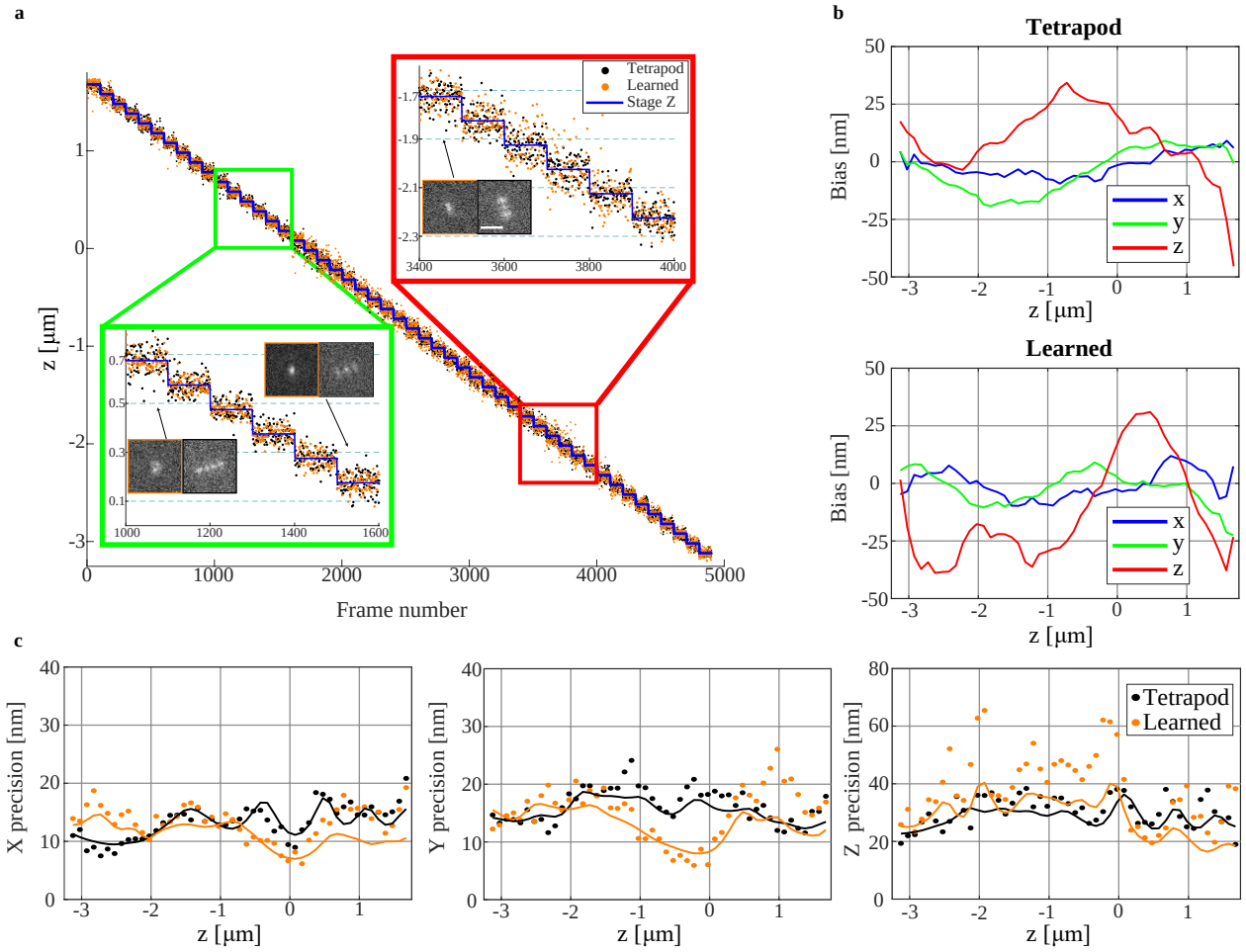


Fig. SN10.5. Experimental precision at low SNR. **a** CNN localizations for both the tetrapod (black dots) and the learned (orange dots) PSFs overlaid on top of the stage readout position (blue steps). each z position was localized with 100 different noise realizations. Example PSFs are shown at both the lower (green inset) and the higher (red inset) parts of the axial range. **b** calibrated bias according to the stage readout for both the Tetrapod (top) and the Learned (bottom) PSFs. **c** Calibrated precision in all three axis for both PSFs. dots mark the estimated precision and continuous lines mark the calculated CRLB. Scale bar in red inset in **a** is 2 μm.

10.5 STORM simulation

To truely compare the performance of the learned PSF to the Tetrapod PSF in a STORM experiment, we need to match the imaged structure, observed emitter blinking and SNR, and have access to GT positions. Since all of this is practically impossible to control experimentally, we performed this comparison in simulation. To make our simulation as realistic as possible, we adopted the GT structure and SNR from the EPFL high density training set MT0N1HD [4] (Fig. SN10.6a). This dataset is composed of 3 interleaving microtubules, imaged over 2500 frames of highly dense emitters in 3D. The axial range of the structure was stretched by a factor of $\frac{1}{4}$ to cover $\approx 3.5\ \mu$. The lateral positions of the emitters were stretched by a factor of $\frac{110nm}{100nm} = 1.1$ to match the CCD pixel size that the learned PSF was designed for. As for the SNR, we empirically matched it to the EPFL simulation using our noise model. To achieve this, we multiplied the number of photons from the EPFL simulation by 2 to turn them into counts. Afterwards, these counts were assumed to follow a Poisson distribution in addition to uniform background of 20 counts per pixel. Finally, the resulting frames were also corrupted by a read noise (assumed to be Gaussian) with a standard deviation of $\sigma = 10$ counts. Example two resulting frames are shown in Fig. SN10.6b.

The results we got in terms of the Jaccard index, lateral RMSE, and axial RMSE, were 40% (43.5%), 31 nm (28 nm), and 31 nm (32.5 nm), for the Tetrapod (learned) PSF respectively. This means the learned PSF improved 3.5 % in terms of detection, gained ≈ 3 nm in lateral resolution, and lost ≈ 1.5 nm in axial resolution. While these differences might feel insignificant, the resulting reconstructions prove otherwise (Fig. SN10.6c,d). For example, the combination of the increased detection with the higher lateral precision is quite apparent in the resulting lateral resolution (SN10.6c inset (i)). Moreover, the smaller lateral footprint of the learned PSF enabled detection of emitters nearby the edge of the FOV which were missed with the Tetrapod ((SN10.6c inset (ii)). Furthermore, despite the $\approx$1.5 decrease in the axial RMSE, the recovered crossing of the microtubules in the XZ cross-section ((SN10.6c inset (iii)) was more accurate with the learned PSF due to the increased detection.

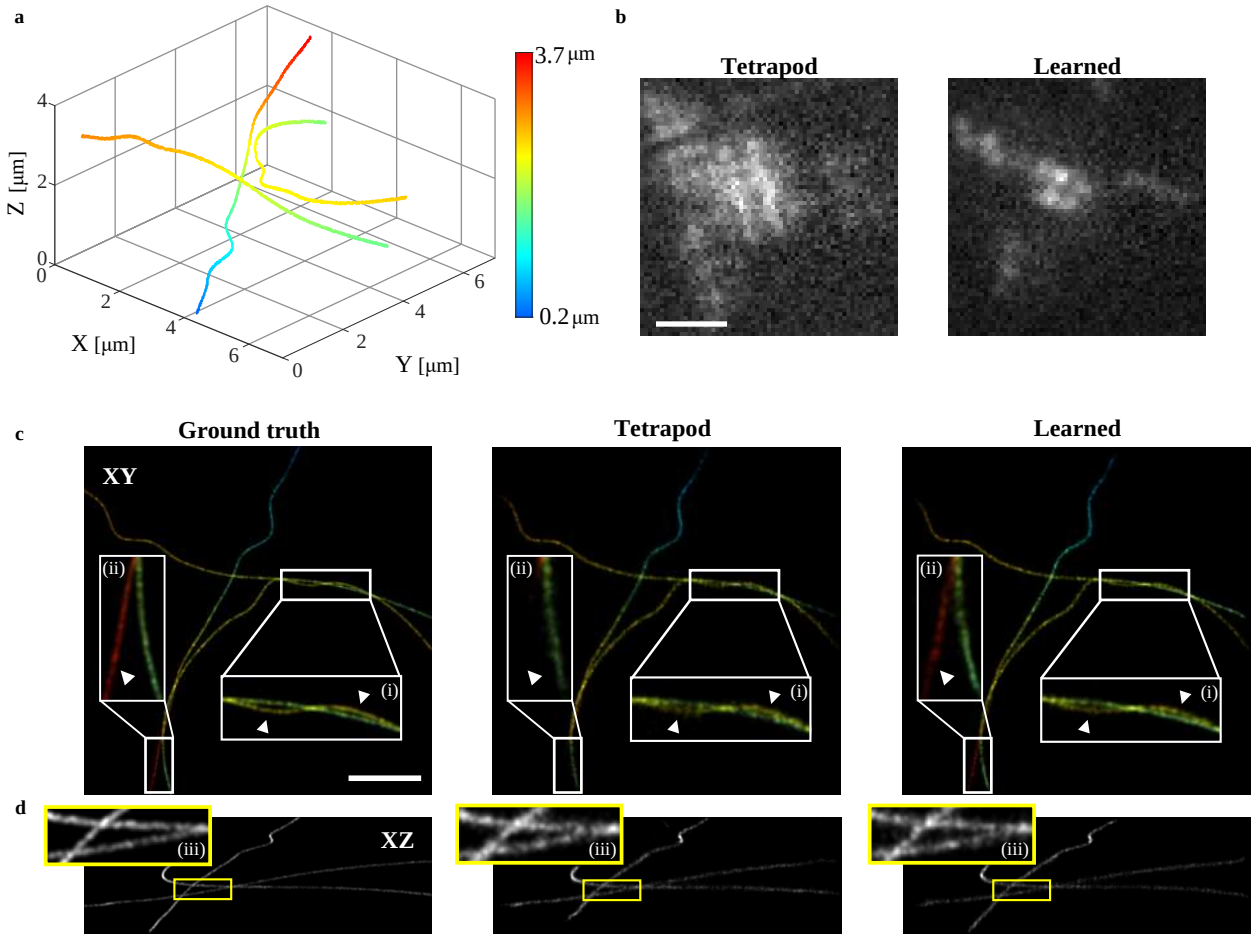


Fig. SN10.6. Simulated STORM experiment. **a** GT 3D structure where z is encoded by color. **b** Example frame with 17 emitters encoded with the Tetrapod (left) and the Learned (right) PSFs. **c** 3D histograms with an isotropic 10 nm grid, and blurred with a gaussian with a standard deviation of 15 nm for the GT strcuture (left), the Tetrapod PSF (middle), and the learned PSF(right). **d** XZ cross-sections of the corresponding histogram in **c**. Scale bars are 2 μm.

Finally, while our learned PSF proved to be superior to the Tetrapod PSF also in STORM conditions, keep in mind that it was designed for the telomere conditions, namely axial range of $[2, 6]$ and High SNR. Therefore, to truely optimize the performance with our learned PSF for STORM imaging, we initialized the phase mask to the learned PSF for the telomere conditions (Fig. SN10.7a top), decreased the SNR to match the STORM conditions, and increased the density by a factor of $\times 2$ (Fig. SN10.7a middle) and factor of $\times 3$ (Fig. SN10.7a bottom).

Interestingly, at low SNR the resulting PSF was simply a faster revolving version of the high SNR PSF in order to account for the axial range shrinkage due to refractive index mismatch ($[0, 4]$ in STORM compared to $[2, 6]$ for telomeres). This effectively improved the expected axial localization precision (Fig. SN10.7b right panel), allowing it to reach the performance limit of the Tetrapod even in the single emitter case. Moreover, the increased density (Fig. SN10.7a middle and bottom panels) didn't seem to phase the optical design network, meaning our PSF can work at much higher densities, unlocking new grounds not explored before in dense localization.

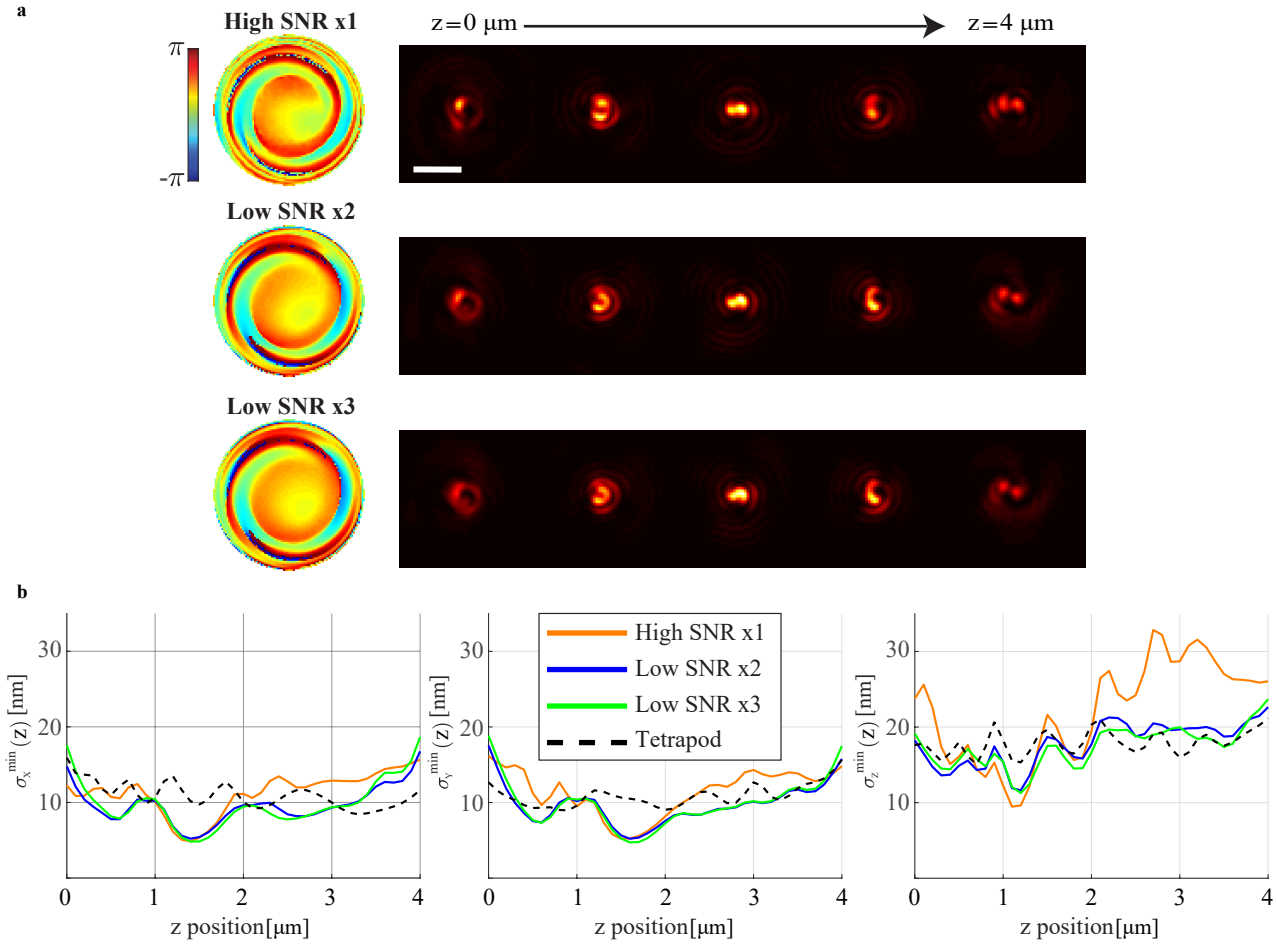


Fig. SN10.7. Mask optimization for STORM imaging. **a** Learned PSF for telomere SNR and density (top), for STORM SNR and $\times 2$ telomere density (middle), and for STORM SNR and $\times 3$ telomere density (bottom). **b** CRLB comparison of the learned PSFs with the Tetrapod. The CRLB was calculated assuming $9K$ signal counts with 160 counts per-pixel background. Similarly to [25] differentiation is done numerically with 1 nm perturbations. Scale bar is 2 μm.

11 Phase retrieval and wobble correction

An accurate PSF model is crucial to achieve optimal localization precision. Hence, to correct for optical aberrations we implemented a Phase Retrieval (PR) algorithm similar to [32, 44]. First, we scanned the objective with 80 nm steps to acquire an axial stack $\left\{y_{f_{nom}^i}\right\}_{i=1}^{80}$ of a single bead (Tetraspeck 0.2um) using the same optical setup of the biological experiments, i.e. excited by a 561 nm laser (Toptica iChrome MLE), filtered (Chroma 575/90 bandpass) to have a similar wavelength to the cell experiments.

Note that since the bead is imaged on the coverslip, the imaging model discussed in section 3.1 is not a perfect representation. This is because, near the coverslip (e.g. < 1 μm) we need to account for super-critical angle fluorescence which the mask wasn't designed for. Therefore, to nullify this model mismatch from the PR process, and exclusively capture optical aberrations that will be present deeper in the sample, we used a vectorial diffraction model that assumes freely rotating dipoles [25].

The axial position of the bead was fixed to 0.1 μm which is the bead radius. To calibrate the wobble as function of the axial position, we define the centroid of the axial slice matching the focus setting $f_{nom} = 0$ to be the origin $(x_0, y_0) = (0, 0)$. Moreover, the additive aberration was assumed to be a combination of the first 50 Zernike polynomials not including piston and tilt:

$$M_{\text{retrieved}} = M_{\text{theory}} + \sum_{j=2}^{50} a_j Z_j \tag{S40}$$

This assumption simplifies the optimization process greatly, and reduces it to estimating only 48 Zernike coefficients. On the other hand, this computational relief comes at the cost of modelling capacity since Zernike polynomials are smooth functions and not well fitted to model phase-jumps (Fig. SN11.1 a (right panel)) such as in the learned mask or the double helix mask [1]. Nevertheless, we were able to obtain excellent results with the theoretical learned mask, therefore, given the only approximate experimental GT, we used PR only to refine the Tetrapod mask.

Next, let $M_{\text{retrieved}}$ denote the retrieved phase mask, $y_{f_{nom}^i}$ denote the PSF image at focus position f_{nom}^i, and (x_i, y_i) denote the lateral displacement from the defined origin. The PR algorithm alternates between two steps:

1. Fix the retrieved phase mask $M_{\text{retrieved}}$, and use MLE in conjunction with the model (equation (S13)) to estimate the focus position f_{nom}^i, the SNR (N_i, b_i), and the wobble (x_i, y_i) in each axial slice $y_{f_{nom}^i}$:

$$\hat{\Theta}_i = \left(\hat{x}_i, \hat{y}_i, \hat{f}_{nom}^i, \hat{N}_i, \hat{b}_i\right) = \underset{\Theta_i}{\text{argmin}} \left(-log\left(\mathcal{L}\left(\Theta; y_{f_{nom}^i}\right)\right)\right) \quad \forall i \in \{1, ..., 80\} \tag{S41}$$

2. Fix $\{\Theta_i\}_{i=1}^{80}$, calculate the respective model images $\left\{I_{\hat{\Theta}_i}\right\}_{i=1}^{80}$, and update the retrieved phase mask $M_{\text{retrieved}}$:

$$\hat{a}_j = \underset{a_j}{\text{argmin}} \sum_{t=1}^{T} \sum_{s=1}^{S} \sum_{i=1}^{80} \left| I_{\hat{\Theta}_i}[t, s] - y_{\hat{f}_{nom}^i}[t, s] \right| \quad \forall j \in \{2, ..., 50\}$$

$$M_{\text{retrieved}} = M_{\text{theory}} + \sum_{j=2}^{50} \hat{a}_j Z_j \tag{S42}$$

The retrieved phase mask $M_{\text{retrieved}}$ was initialized to the LC-SLM calibrated theoretical mask M_{theory}, and $\left\{\hat{f}_{nom}^i\right\}_{i=1}^{80}$ were initialized to the designed scan positions. Note that differently from [44], here we employed this alternation strategy since the result of each step depends on the result of the other. Therefore, for an accurate calibration of the wobble [78] over a large axial range we need to calibrate it using the already retrieved pupil function. Moreover, this approach was more accurate than simply assuming a known focus position from the stage readout. We found that for our setup 2 iterations were enough to achieve convergence.

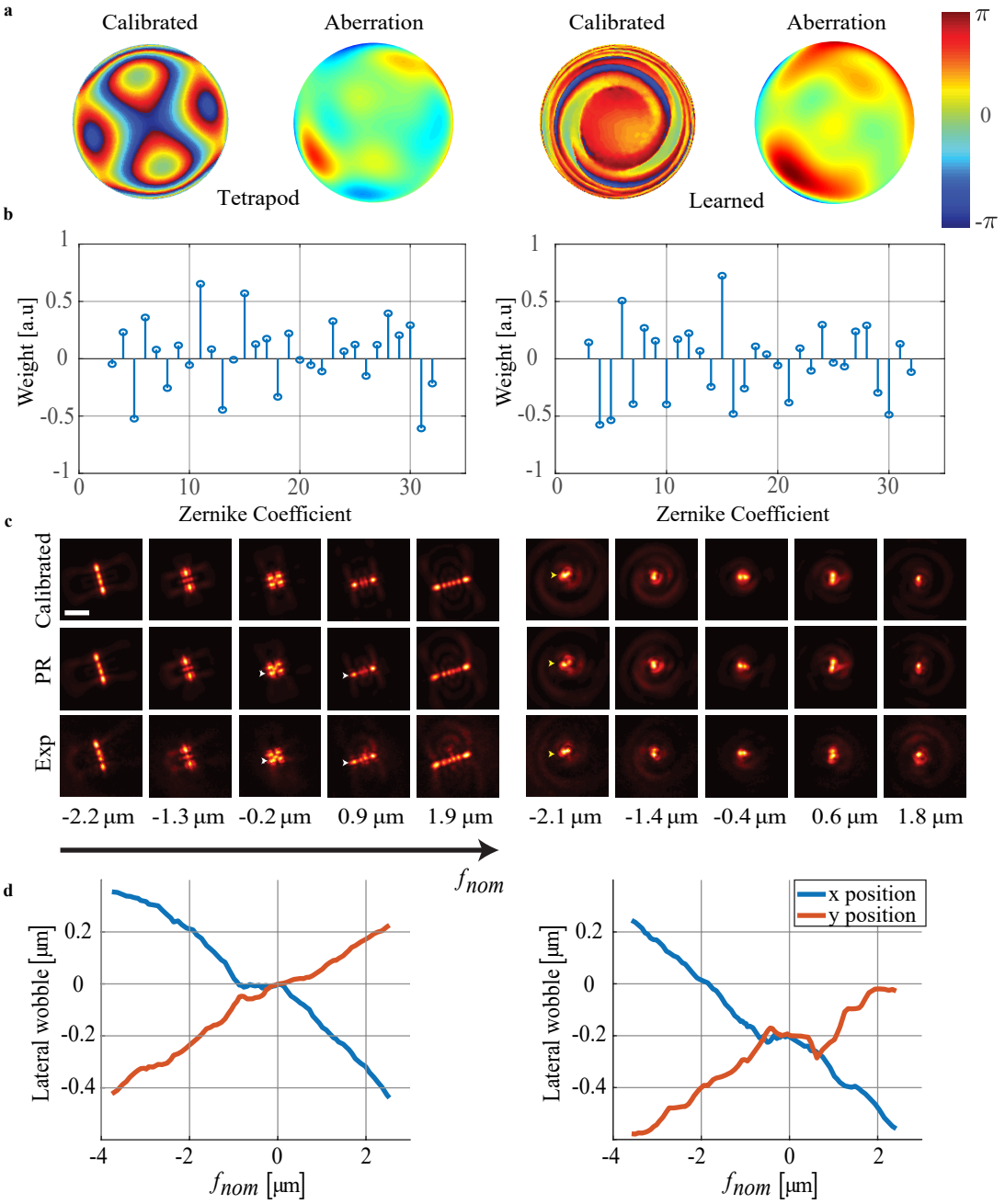


Fig. SN11.1. Phase retrieval and wobble correction. **a** Calibrated mask and aberration for the Tetrapod (left) and the learned (right) PSFs. Calibration is achieved by projecting the desired phase pattern on the available LC-SLM calibration voltages. **b** Corresponding Zernike coefficients (according to Noll indexing) of the abberation in (**a**). The coefficients for higher polynomials were negligible, and therefore omitted from the plot. **c** Comparison of the simulated PSFs using the calibrated/retrieved mask to an experimentally measured z-stack of a fluorescent bead with a similar emission wavelength to the cell data. The PR algorithm managed to recover the aberration for the Tetrapod mask (white arrows), and failed to recover it for the learned mask (yellow arrows). **c** Calibrated lateral wobble as function of the focus position for the Tetrapod (left) and the learned mask (right). Scale bar is 3 μm.

12 Experimental ground truth

To approximate ground truth 3D positions of the telomeres (Fig. SN12.1), we scanned the sample in the axial direction with 100 nm steps covering a 5 μm range (see Supplementary Video 5). Next, the telomeres were localized in each frame using ThunderSTORM [79] to extract the lateral position (i.e. XY centroid). Afterwards, the approximate axial position of each detected telomere was determined by fitting a 2^{nd} order polynomial to the mean intensity profile along 17 adjacent axial slices (Fig. SN12.1 b-d). The resulting z locations were multiplied by a factor of $\frac{1.33}{1.518}$ to account for refractive index mismatch [80, 81]. To compare the recovered positions to the approximate experimentally calibrated GT, we corrected the lateral recovered position using the wobble calibration matching the recovered axial position. To estimate the wobble for unmeasured axial positions we used cubic spline interpolation.

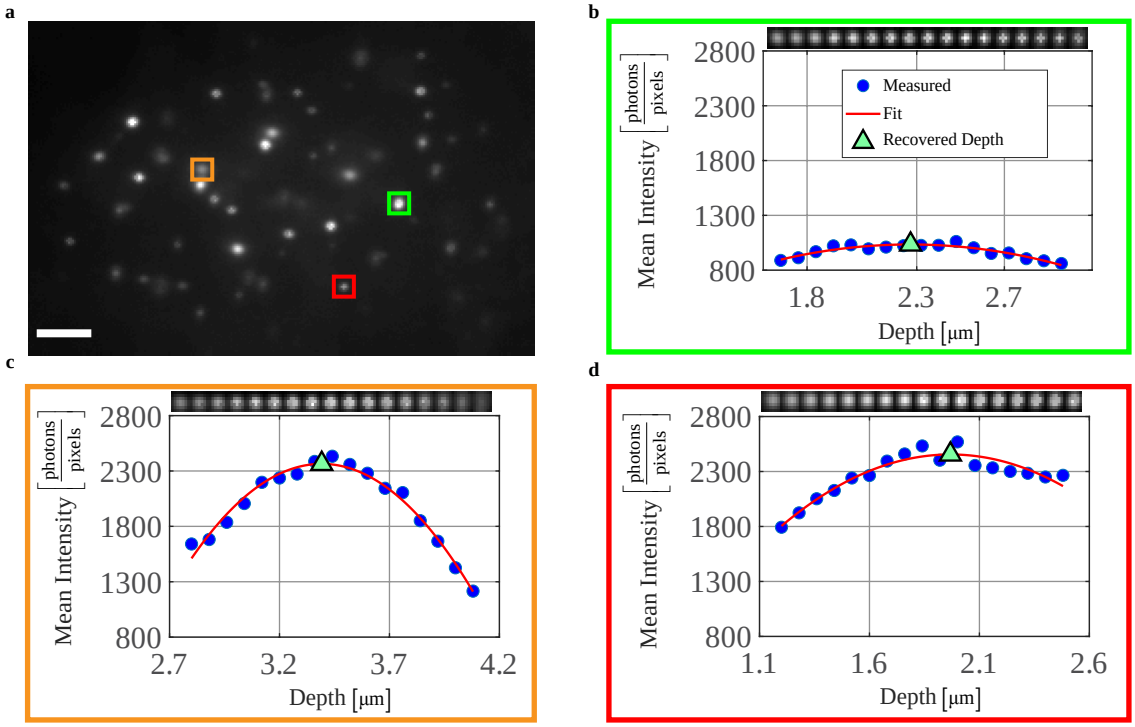


Fig. SN12.1. Experimental ground truth estimation. **a** Focus slice with 3 marked emitters. **b - d** Estimation of the axial position for the 3 emitters. The emitters vary in size (e.g. **b** vs. **d**) and signal counts (e.g. **b** vs. **c**), therefore the fit accuracy is limited. Scale bar is 3 μm.

13 Telomere imaging

13.1 Additional fixed cell results

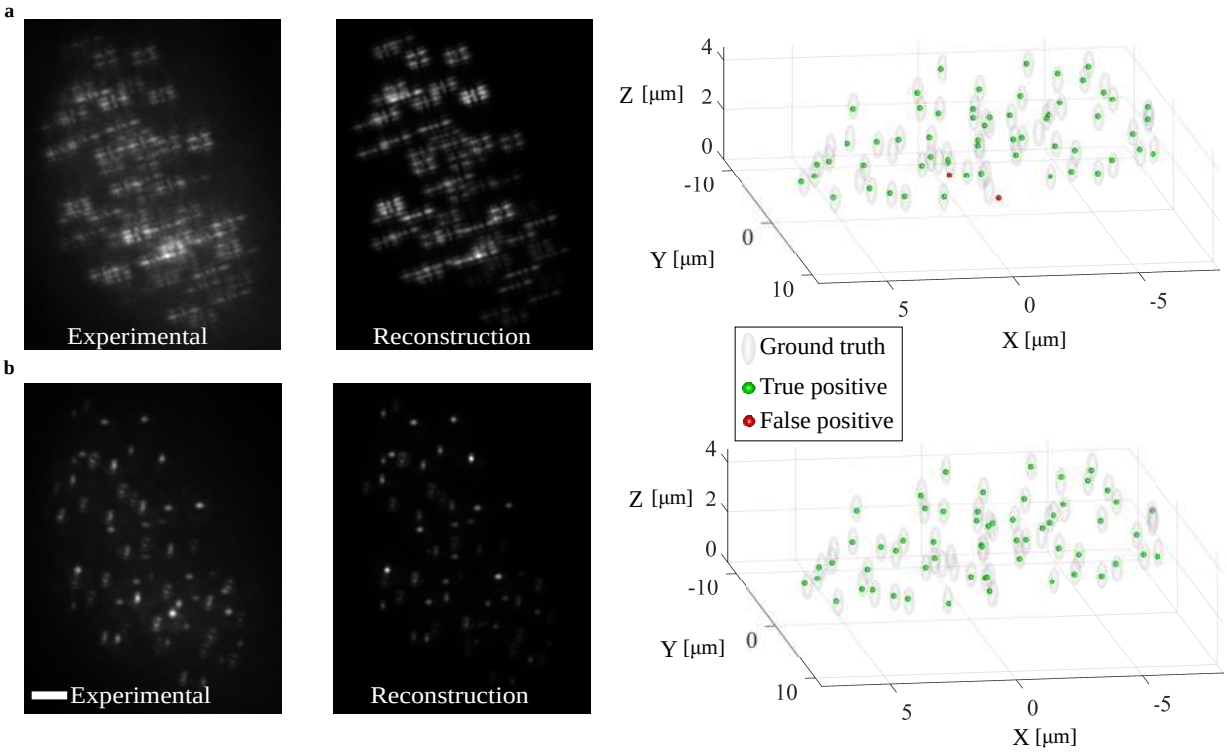


Fig. SN13.1. Experimental demonstration for a higher focus setting. **a** Experimental snapshot with the Tetrapod PSF (left), rendered image from the 3D recovered positions by the Tetrapod CNN (middle), and a 3D comparison of the recovered positions and the approximate experimental ground truth (right). **b** Experimental snapshot with the learned PSF (left), rendered image from the 3D recovered positions by the learned PSF CNN (middle), and a 3D comparison of the recovered positions and the approximate experimental ground truth (right). Note that the reconstructions PSFs were scaled according to their retrieved intensity, therefore some appear dim, however their positions are correctly recovered as apparent in the right figures. The Jaccard index for the Tetrapod PSF was 0.85 compared to 0.89 for the learned PSF. Scale bar is 3 μm.

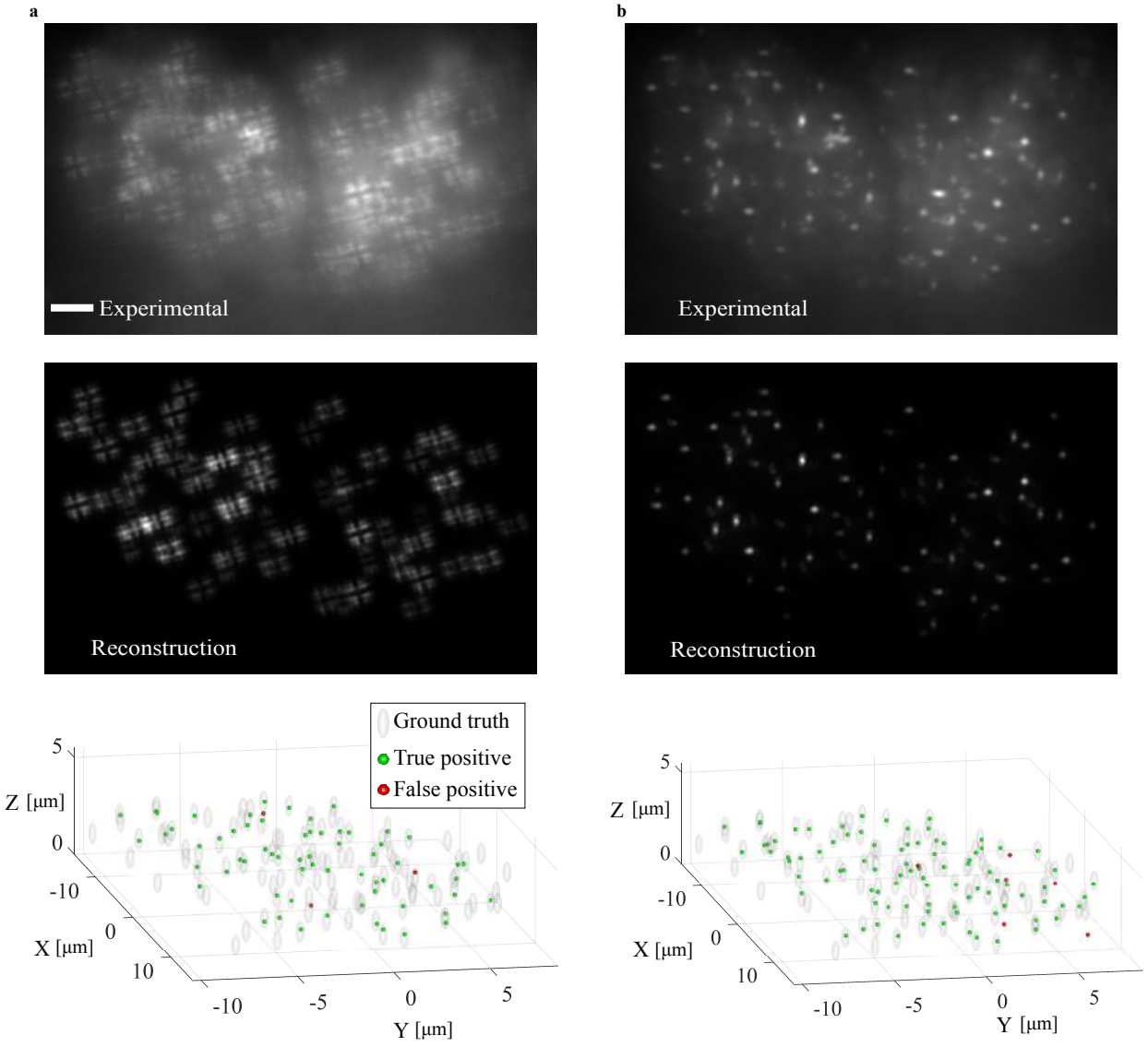


Fig. SN13.2. Experimental demonstration for a lower SNR. **a** Experimental snapshot with the Tetrapod PSF (top), rendered image from the 3D recovered positions by the Tetrapod CNN (middle), and a 3D comparison of the recovered positions and the approximate experimental ground truth (bottom). **b** Experimental snapshot with the learned PSF (top), rendered image from the 3D recovered positions by the learned PSF CNN (middle), and a 3D comparison of the recovered positions and the approximate experimental ground truth (bottom). Note that the reconstructions PSFs were scaled according to their retrieved intensity, therefore some appear dim, however their positions are correctly recovered as apparent in the right figures. The Jaccard index for the Tetrapod PSF was 0.52 compared to 0.72 for the learned PSF. Scale bar is 3 μm.

13.2 Image normalization

To cope with the dramatically decreasing SNR throughout the live cell imaging experiment, we stretched the contrast of each training/testing frame to the range $[0, 1]$ (Fig. SN13.3) using the transformation:

$$I_{[0,1]} = \frac{I - I_{min}}{I_{max} - I_{min}} \tag{S43}$$

Where I_{min} and I_{max} are the minimum and maximum pixel value in frame I. This ensures that pixel values in the transformed image lie within a narrow range(Fig. SN13.3) which enables learning a single network to localize the entire recorded movie.

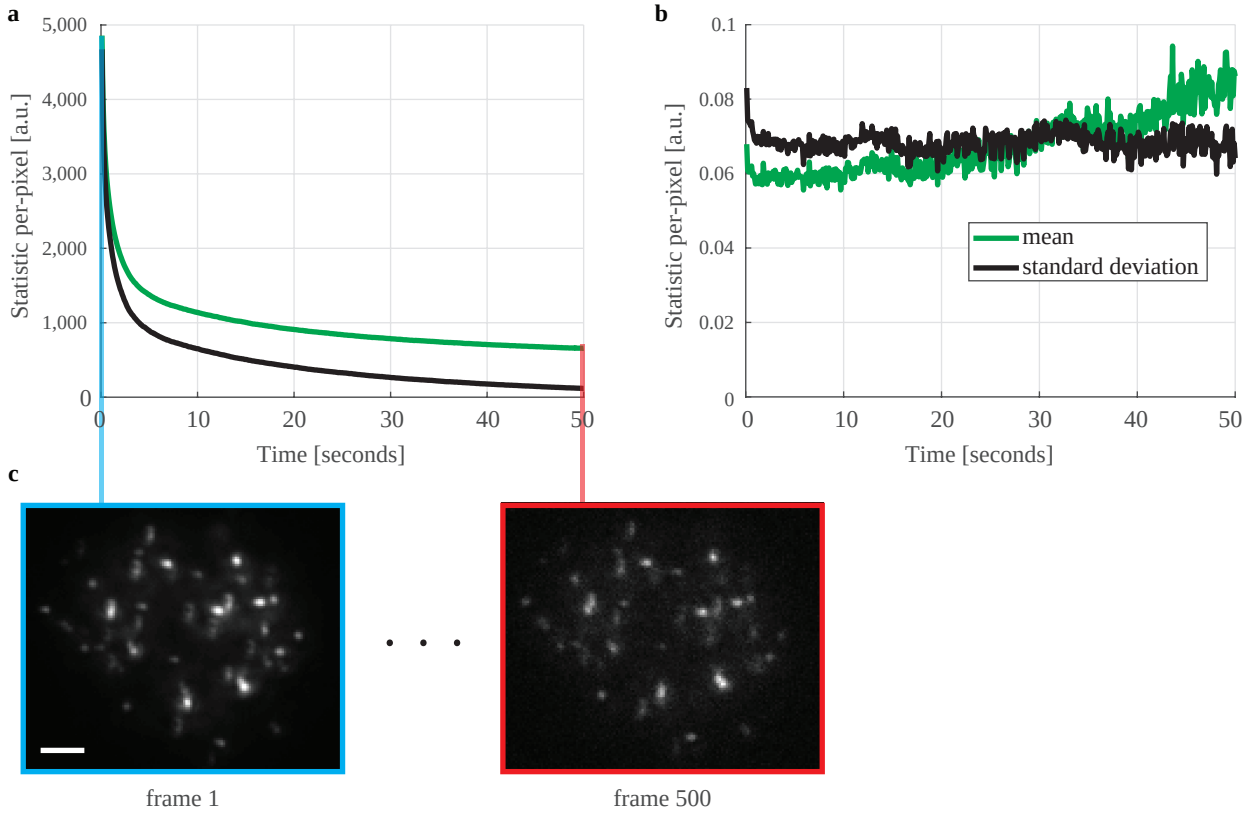
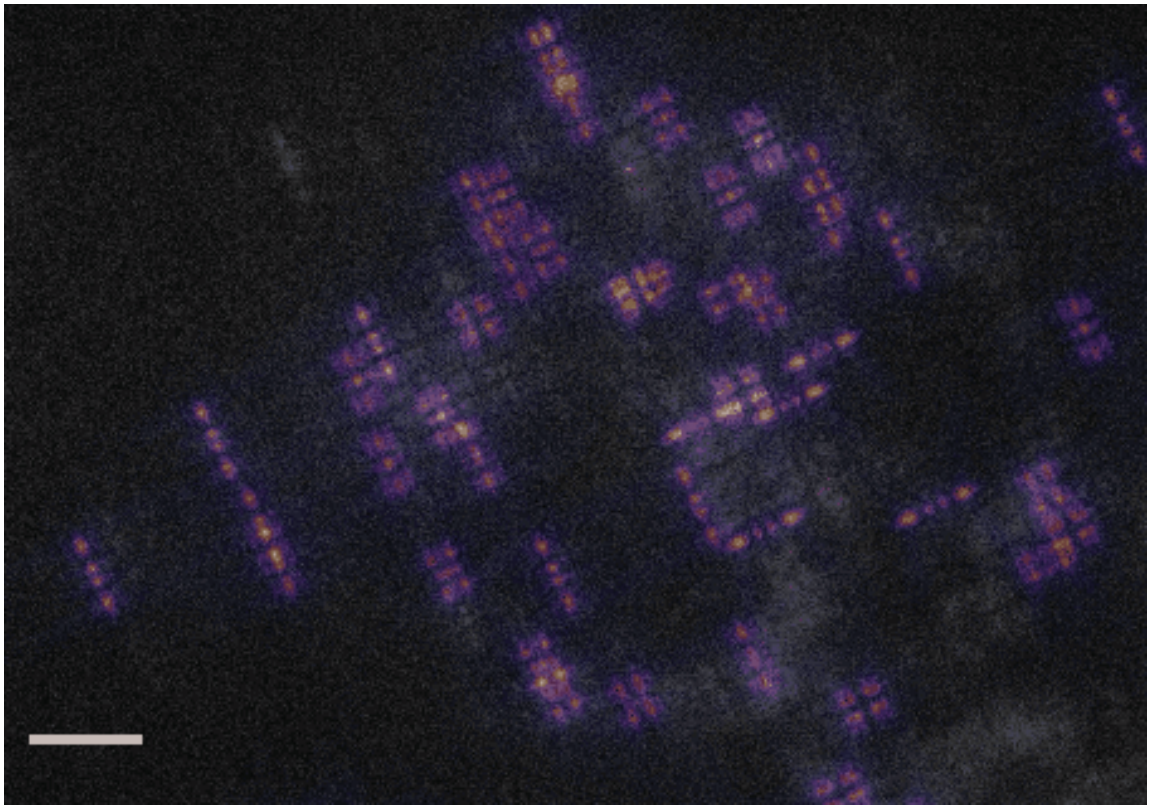


Fig. SN13.3. Normalization to the range $[0, 1]$. **a** Mean and standard deviation of experimentally recorded counts per pixel as function of time. The number of pixels in each experimental frame was $n = 15,723$. **b** Mean and standard deviation of the transformed experimental pixel values. **c** First (left) and last (right) experimental frames in the recorded movie. Scale bar is 3 μm.

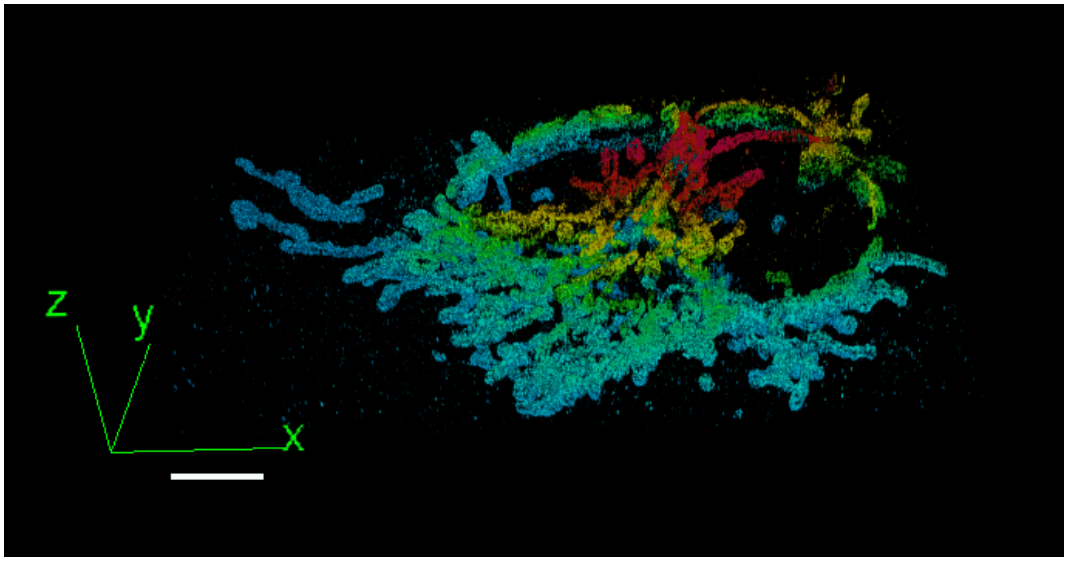
13.3 Track linking and post-processing

The per-frame localizations were linked based on a simple 3D proximity tracker. All tracks started within the first 7 frames and were relatively clustered in 3D with no bifurcations observed (see Suplementary Video 7). After linking, all tracks were smoothed using a moving average filter of length 0.5 s (5 frames). Of course, mean squared displacement calculations were performed on the raw tracks prior to smoothing. Finally, for more complicated tracking scenarios the reader is encouraged to link the CNN localizations by resorting to a more robust tracking software such as [82].

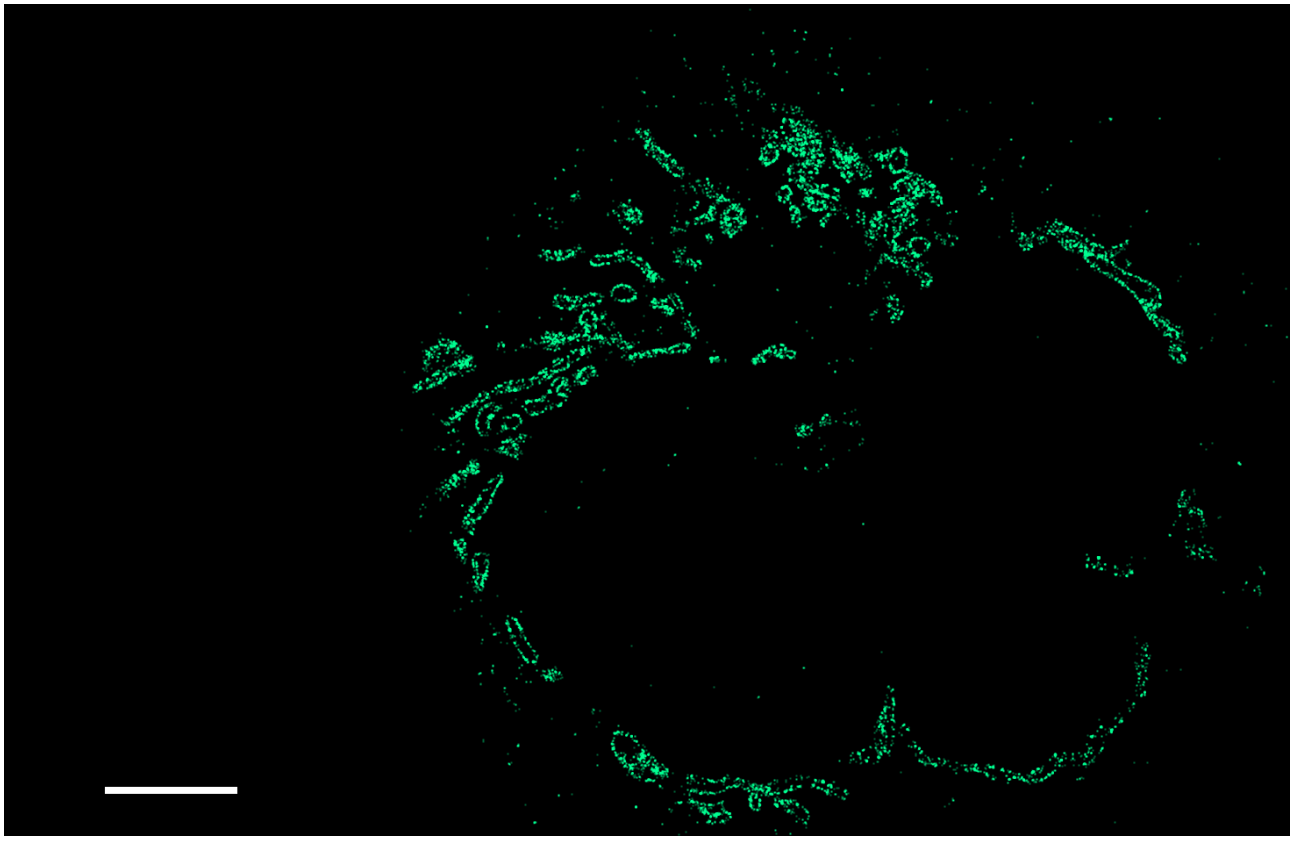
14 Supplementary videos



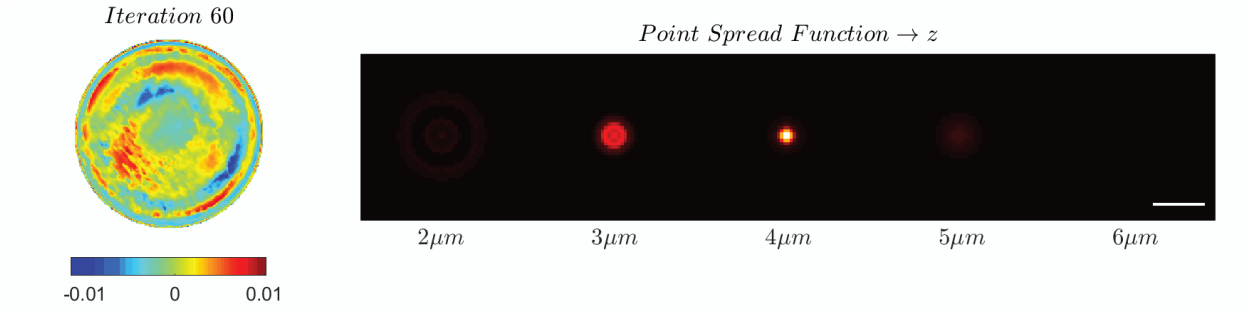
Supplementary Video 1. Localizations overlaid on experimental frames. This movie shows 70 representative experimental frames followed by an overlay of their re-generated images using the recovered 3D positions by the CNN (Fig. 3b main text). Note that the experimental frames are shown before and after the re-generated images for ease of visualization. The STORM experiment was repeated independently for $n = 3$ cells, twice analyzing 20K frames and once analyzing 10K frames all leading to similar performance. Scale bar is 5 μm.



Supplementary Video 2. Rotating 3D rendering of the recovered mitochondria. This movie shows a 3D rendering of the super-resolved mitochondria spanning a 4 μm axial range (Fig. 3a main text). The z-range is rendered with a scaling factor of 2 to ease axial visualization. Scale bar is 5 μm.

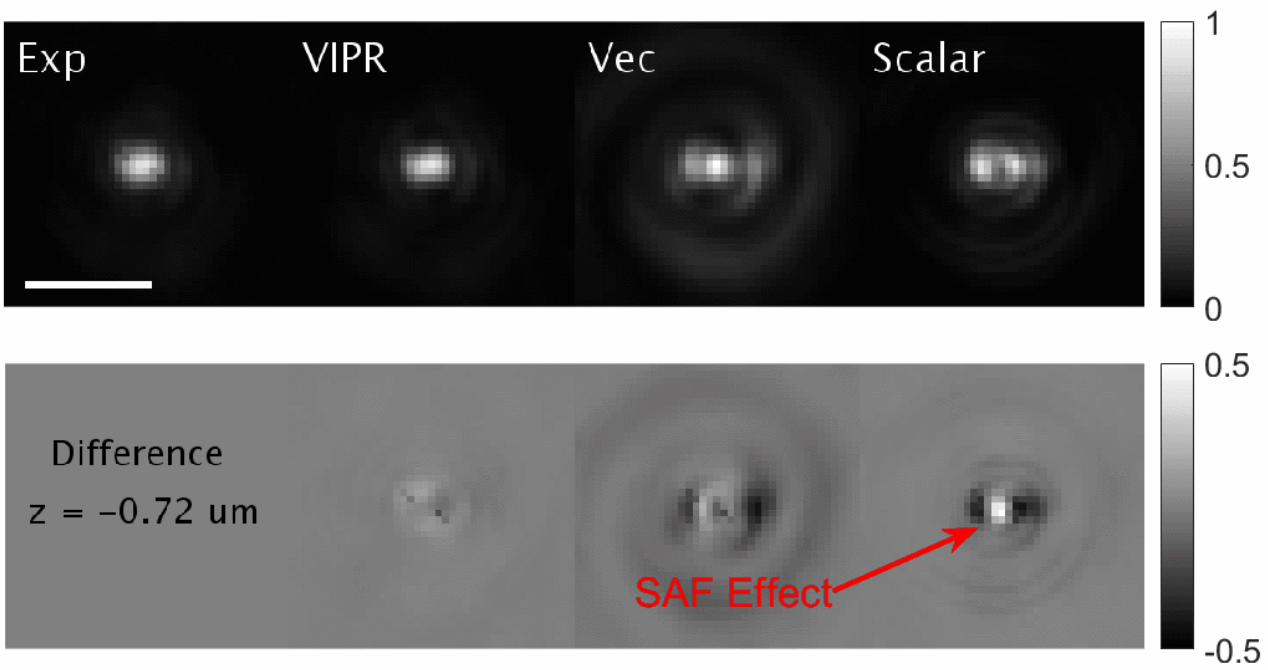


Supplementary Video 3. Sweep through the axial slices of the recovered mitochondria. This movie shows a sweep through 33 nm axial slices of the rendered 3D histogram for the mitochondria data (Fig. 3a main text). Scale bar is 5 μm.

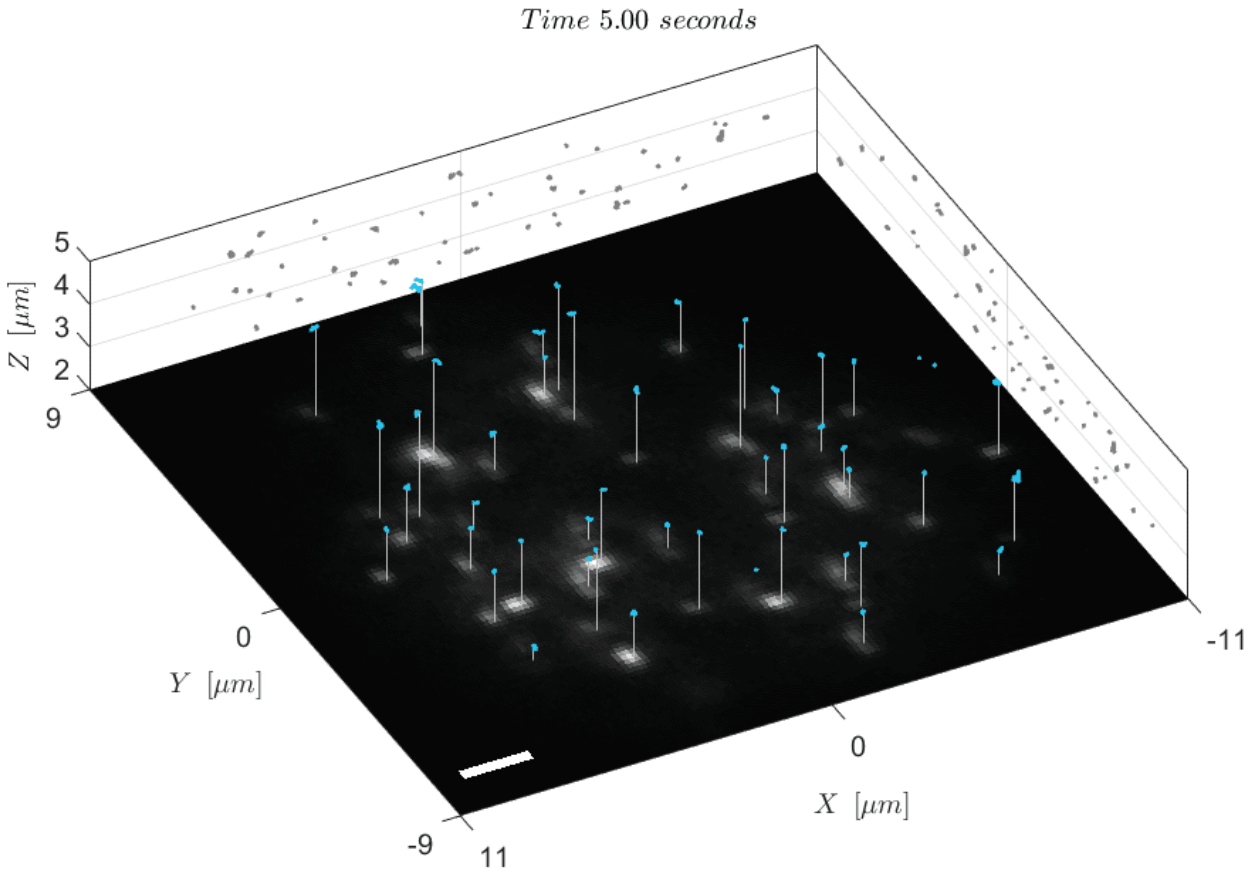


Supplementary Video 4. Phase mask learning via backpropagation. This movie shows the phase mask (left) and the corresponding PSF (right) being learned over training iterations (Fig. 4c main text). Note that the phase mask is initialized to zero modulation, meaning the standard microscope PSF. Scale bar is 2μm.

Supplementary Video 5. Rotating Telomere z-stack without a mask. This movie shows a 3D rendering of the telomere data z-stack without the application of a phase mask (Fig. 5b main text). As clearly shown in the rendered PSFs, the telomeres exhibit different sizes and intensities. The experiment was repeated independently for $n = 10$ U2OS cells all showing similar characteristics. Scale bar is 5 μm.



Supplementary Video 6. SAF light effect on the learned PSF. This movie shows the effect of the SAF light on the experimental PSF with the learned phase mask. Upper panel shows the experimental PSF (left), the result of VIPR [46], the vectorial model assuming dipole emission, and the scalar model assuming isotropic emission. The lower panel shows the difference from the experimental measurement for each model. The SAF light effect is indicated in the middle of the axial range with a red arrow. Scale bar is 2 μm.



Supplementary Video 7. Volumetric tracking of telomeres in MEF cells. This movie shows 3D tracking of telomeres in live MEF cells over a period of 50 seconds using the learned phase mask (Fig. 6a main text). White sticks point to the emitter being tracked. Time is encoded in color. The results indicate that individual telomeres exhibit different types of movements. The experiment was repeated independently for $n = 10$ MEF cells all showing similar characteristics and performance. Scale bar is 2 μm.

References

1. Pavani, S. R. P. *et al.* Three-dimensional, single-molecule fluorescence imaging beyond the diffraction limit by using a double-helix point spread function. *Proceedings of the National Academy of Sciences* **106,** 2995–2999 (2009).
2. Shechtman, Y., Sahl, S. J., Backer, A. S. & Moerner, W. Optimal point spread function design for 3D imaging. *Physical review letters* **113,** 133902 (2014).
3. Shechtman, Y., Weiss, L. E., Backer, A. S., Sahl, S. J. & Moerner, W. Precise three-dimensional scan-free multiple-particle tracking over large axial ranges with tetrapod point spread functions. *Nano letters* **15,** 4194–4199 (2015).
4. Sage, D. *et al.* Super-resolution fight club: assessment of 2D and 3D single-molecule localization microscopy software. *Nature methods* **16,** 387 (2019).
5. Antipa, N. *et al.* DiffuserCam: lensless single-exposure 3D imaging. *Optica* **5,** 1–9 (2018).
6. Long, J., Shelhamer, E. & Darrell, T. *Fully convolutional networks for semantic segmentation* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), 3431–3440.
7. Ioffe, S. & Szegedy, C. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift* in *Proceedings of the 32nd International Conference on Machine Learning* (eds Bach, F. & Blei, D.) **37** (PMLR, Lille, France, 2015), 448–456.
8. Ulyanov, D., Vedaldi, A. & Lempitsky, V. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022* (2016).
9. Yu, F. & Koltun, V. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122* (2015).
10. He, K., Zhang, X., Ren, S. & Sun, J. *Deep residual learning for image recognition* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), 770–778.
11. Newell, A., Yang, K. & Deng, J. *Stacked hourglass networks for human pose estimation* in *European Conference on Computer Vision* (2016), 483–499.

12. Pavlakos, G., Zhou, X., Derpanis, K. G. & Daniilidis, K. *Coarse-to-fine volumetric prediction for single-image 3D human pose* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), 7025–7034.
13. Odena, A., Dumoulin, V. & Olah, C. Deconvolution and checkerboard artifacts. *Distill* **1,** e3 (2016).
14. Zeiler, M. D., Krishnan, D., Taylor, G. W. & Fergus, R. *Deconvolutional networks* in *2010 IEEE Computer Society Conference on computer vision and pattern recognition* (2010), 2528–2535.
15. Zeiler, M. D. & Fergus, R. *Visualizing and understanding convolutional networks* in *European conference on computer vision* (2014), 818–833.
16. Dumoulin, V. & Visin, F. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285* (2016).
17. Shi, W. *et al. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), 1874–1883.
18. Aitken, A. *et al.* Checkerboard artifact free sub-pixel convolution: A note on sub-pixel convolution, resize convolution and convolution resize. *arXiv preprint arXiv:1707.02937* (2017).
19. Maas, A. L., Hannun, A. Y. & Ng, A. Y. *Rectifier nonlinearities improve neural network acoustic models* in *Proc. icml* **30** (2013), 3.
20. Zhang, H., Li, Q. & Sun, Z. *Joint Voxel and Coordinate Regression for Accurate 3D Facial Landmark Localization* in *2018 24th International Conference on Pattern Recognition (ICPR)* (2018), 2202–2208.
21. LeNail, A. NN-SVG: Publication-ready neural network architecture schematics. *Journal of Open Source Software* **4,** 747 (2019).
22. Haim, H., Elmalem, S., Giryes, R., Bronstein, A. M. & Marom, E. Depth estimation from a single image using deep learned phase coded mask. *IEEE Transactions on Computational Imaging* **4,** 298–310 (2018).
23. Sitzmann, V. *et al.* End-to-end optimization of optics and image processing for achromatic extended depth of field and super-resolution imaging. *ACM Transactions on Graphics (TOG)* **37,** 114 (2018).
24. Wu, Y., Boominathan, V., Chen, H., Sankaranarayanan, A. & Veeraraghavan, A. *PhaseCam3D—Learning Phase Masks for Passive Single View Depth Estimation* in *2019 IEEE International Conference on Computational Photography (ICCP)* (2019), 1–12.
25. Backer, A. S. & Moerner, W. Extending single-molecule microscopy using optical Fourier processing. *The Journal of Physical Chemistry B* **118,** 8313–8329 (2014).
26. Goodman, J. W. *Introduction to Fourier optics* (Roberts and Company Publishers, 2005).
27. Liu, S., Kromann, E. B., Krueger, W. D., Bewersdorf, J. & Lidke, K. A. Three dimensional single molecule localization using a phase retrieved pupil function. *Optics express* **21,** 29462–29487 (2013).
28. Bourg, N. *et al.* Direct optical nanoscopy with axially localized detection. *Nature Photonics* **9,** 587 (2015).
29. Richards, B & Wolf, E. Electromagnetic diffraction in optical systems, II. Structure of the image field in an aplanatic system. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* **253,** 358–379 (1959).
30. Boyd, N., Jonas, E., Babcock, H. P. & Recht, B. Deeploco: Fast 3D localization microscopy using neural networks. *BioRxiv,* 267096 (2018).
31. Babcock, H. P. & Zhuang, X. Analyzing single molecule localization microscopy data using cubic splines. *Scientific reports* **7,** 552 (2017).
32. Petrov, P. N., Shechtman, Y. & Moerner, W. Measurement-based estimation of global pupil functions in 3D localization microscopy. *Optics express* **25,** 7945–7959 (2017).
33. Aristov, A., Lelandais, B., Rensen, E. & Zimmer, C. ZOLA-3D allows flexible 3D localization microscopy over an adjustable axial range. *Nature communications* **9,** 2409 (2018).
34. Hirsch, M., Wareham, R. J., Martin-Fernandez, M. L., Hobson, M. P. & Rolfe, D. J. A stochastic model for electron multiplication charge-coupled devices–from theory to practice. *PloS one* **8,** e53671 (2013).
35. Huang, F. *et al.* Video-rate nanoscopy using sCMOS camera–specific single-molecule localization algorithms. *Nature methods* **10,** 653 (2013).
36. Kingma, D. P. & Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).
37. Remmert, R. *Theory of complex functions* (Springer Science & Business Media, 2012).
38. Al-Rfou, R. *et al.* Theano: A Python framework for fast computation of mathematical expressions. *arXiv preprint arXiv:1605.02688* (2016).
39. Kreutz-Delgado, K. The complex gradient operator and the CR-calculus. *arXiv preprint arXiv:0906.4835* (2009).
40. Paszke, A. *et al.* Automatic differentiation in pytorch (2017).
41. Rippel, O., Snoek, J. & Adams, R. P. *Spectral representations for convolutional neural networks* in *Advances in neural information processing systems* (2015), 2449–2457.
42. Li, Y. *et al.* Real-time 3D single-molecule localization using experimental point spread functions. *Nature methods* **15,** 367 (2018).
43. Born, M. & Wolf, E. *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light* (Elsevier, 2013).

44. Zelger, P *et al.* Three-dimensional localization microscopy using deep learning. *Optics express* **26,** 33166–33179 (2018).
45. Wang, W. *et al.* Generalized method to design phase masks for 3D super-resolution microscopy. *Optics express* **27,** 3799–3816 (2019).
46. Ferdman, B. *et al.* VIPR: Vectorial Implementation of Phase Retrieval for fast and accurate microscopic pixel-wise pupil estimation. *bioRxiv* (2020).
47. Cao, X., Wei, Y., Wen, F. & Sun, J. Face alignment by explicit shape regression. *International Journal of Computer Vision* **107,** 177–190 (2014).
48. Nehme, E., Weiss, L. E., Michaeli, T. & Shechtman, Y. Deep-STORM: super-resolution single-molecule microscopy by deep learning. *Optica* **5,** 458–464 (2018).
49. Nibali, A., He, Z., Morgan, S. & Prendergast, L. Numerical coordinate regression with convolutional neural networks. *arXiv preprint arXiv:1801.07372* (2018).
50. Lin, M., Chen, Q. & Yan, S. Network in network. *arXiv preprint arXiv:1312.4400* (2013).
51. Hershko, E., Weiss, L. E., Michaeli, T. & Shechtman, Y. Multicolor localization microscopy and point-spread-function engineering by deep learning. *Optics express* **27,** 6158–6183 (2019).
52. Lin, T.-Y., Goyal, P., Girshick, R., He, K. & Dollár, P. *Focal loss for dense object detection* in *Proceedings of the IEEE international conference on computer vision* (2017), 2980–2988.
53. Zhang, N., Shelhamer, E., Gao, Y. & Darrell, T. Fine-grained pose prediction, normalization, and recognition. *arXiv preprint arXiv:1511.07063* (2015).
54. Pishchulin, L. *et al. Deepcut: Joint subset partition and labeling for multi person pose estimation* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), 4929–4937.
55. Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M. & Schiele, B. *Deepercut: A deeper, stronger, and faster multi-person pose estimation model* in *European Conference on Computer Vision* (2016), 34–50.
56. Rahman, M. A. & Wang, Y. *Optimizing intersection-over-union in deep neural networks for image segmentation* in *International symposium on visual computing* (2016), 234–244.
57. Milletari, F., Navab, N. & Ahmadi, S.-A. *V-net: Fully convolutional neural networks for volumetric medical image segmentation* in *2016 Fourth International Conference on 3D Vision (3DV)* (2016), 565–571.
58. Lguensat, R. *et al. EddyNet: A deep neural network for pixel-wise classification of oceanic eddies* in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium* (2018), 1764–1767.
59. Berman, M., Rannen Triki, A. & Blaschko, M. B. *The Lovász-Softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), 4413–4421.
60. Luvizon, D. C., Tabia, H. & Picard, D. Human pose regression by combining indirect part detection and contextual information. *arXiv preprint arXiv:1710.02322* (2017).
61. Sun, X., Xiao, B., Wei, F., Liang, S. & Wei, Y. *Integral human pose regression* in *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), 529–545.
62. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
63. Wu, Y. & He, K. *Group normalization* in *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), 3–19.
64. Loshchilov, I. & Hutter, F. Decoupled Weight Decay Regularization (2018).
65. Wan, L., Zeiler, M., Zhang, S., Le Cun, Y. & Fergus, R. *Regularization of neural networks using dropconnect* in *International conference on machine learning* (2013), 1058–1066.
66. Kuhn, H. W. The Hungarian method for the assignment problem. *Naval research logistics quarterly* **2,** 83–97 (1955).
67. Kay, S. M. *Fundamentals of statistical signal processing* (Prentice Hall PTR, 1993).
68. Holden, S. J., Uphoff, S. & Kapanidis, A. N. DAOSTORM: an algorithm for high-density super-resolution microscopy. *Nature methods* **8,** 279 (2011).
69. Huang, F., Schwartz, S. L., Byars, J. M. & Lidke, K. A. Simultaneous multiple-emitter fitting for single molecule super-resolution imaging. *Biomedical optics express* **2,** 1377–1393 (2011).
70. Babcock, H., Sigal, Y. M. & Zhuang, X. A high-density 3D localization algorithm for stochastic optical reconstruction microscopy. *Optical Nanoscopy* **1,** 6 (2012).
71. Abraham, A. V., Ram, S., Chao, J., Ward, E. & Ober, R. J. Quantitative study of single molecule location estimation techniques. *Optics express* **17,** 23352–23373 (2009).
72. Mallat, S. G. & Zhang, Z. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing* **41,** 3397–3415 (1993).

73. Pati, Y. C., Rezaiifar, R. & Krishnaprasad, P. S. *Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition* in *Proceedings of 27th Asilomar conference on signals, systems and computers* (1993), 40–44.

74. Zhang, P. *et al.* Analyzing complex single-molecule emission patterns with deep learning. *Nature methods* **15,** 913 (2018).

75. Descloux, A. C., Grussmayer, K. S. & Radenovic, A. Parameter-free image resolution estimation based on decorrelation analysis. *Nature methods* **16,** 918–924 (2019).

76. Grover, G., Quirin, S., Fiedler, C. & Piestun, R. Photon efficient double-helix PSF microscopy with application to 3D photo-activation localization imaging. *Biomedical optics express* **2,** 3010–3020 (2011).

77. Siemons, M., Hulleman, C., Thorsen, R., Smith, C. & Stallinga, S. High precision wavefront control in point spread function engineering for single emitter localization. *Optics express* **26,** 8397–8416 (2018).

78. Carlini, L., Holden, S. J., Douglass, K. M. & Manley, S. Correction of a depth-dependent lateral distortion in 3D super-resolution imaging. *PLoS One* **10,** e0142949 (2015).

79. Ovesnỳ, M., Křížek, P., Borkovec, J., Švindrych, Z. & Hagen, G. M. ThunderSTORM: a comprehensive ImageJ plug-in for PALM and STORM data analysis and super-resolution imaging. *Bioinformatics* **30,** 2389–2390 (2014).

80. Hell, S, Reiner, G, Cremer, C & Stelzer, E. H. Aberrations in confocal fluorescence microscopy induced by mismatches in refractive index. *Journal of microscopy* **169,** 391–405 (1993).

81. Huang, B., Wang, W., Bates, M. & Zhuang, X. Three-dimensional super-resolution imaging by stochastic optical reconstruction microscopy. *Science* **319,** 810–813 (2008).

82. Tinevez, J.-Y. *et al.* TrackMate: An open and extensible platform for single-particle tracking. *Methods* **115,** 80–90 (2017).