DBlink: Dynamic localization microscopy in super spatiotemporal resolution via deep learning

Supplementary Information

Alon Saguy[1], Onit Alalouf[1], Nadav Opatovski[2], Soohyen Jang[3], Mike Heilemann[3], Yoav Shechtman[1,†]

[1] Department of Biomedical Engineering, Technion – Israel Institute of Technology, Haifa, 3200003, Israel

[2] Russell Berrie Nanotechnology Institute, Technion – Israel Institute of Technology, Haifa, 3200003, Israel

[3] Institute of Physical and Theoretical Chemistry, Goethe-University Frankfurt, 60438, Germany

Contents

1. Simulations

1.1. Filament simulation

The initial ground truth structures were simulated according to the model suggested by Shariff, et al[1]. The simulated ground truth for each video was a binary map containing ones where there was a filament and zeros everywhere else. Then, we applied affine temporally changing transformations to the original structure over predetermined video length (see Table S1). We applied on the initial ground truth structure two types of movements: global shift and global rotation. The shift velocities were chosen from a uniform distribution in the range of [-4, 4] nm per frame, and the rotation velocities were chosen from a uniform distribution in the range of [-3, 3] degrees per frame. Next, we randomly chose the number of blinking events per frame according to a blinking density parameter that states the percentage of the structure that would blink at each frame; namely, the number of pixels that are apparent ('on') in the frame, divided by the total number of pixels that are part of the structure (contain a value of '1' in the ground-truth). We determined the position of each simulated blinking event according to the ground truth structure with additional localization noise randomly chosen from a uniform distribution in the range of [-20, 20] nm. We added additional localizations at random positions in the field of view (FOV) as noise. To simulate motion within a single acquisition frame, we summed the localizations over temporal windows of 10 simulated frames. The temporal window size, together with the blinking density and the movement velocity, sets the maximal temporal resolution of our method. Since there are multiple degrees of freedom in these parameters, we chose to keep the number of summed frames at a constant of 10 frames and to change the emitter density and the velocities in a range that matches our experimental data. The result was pairs of simulated localization video and underlying dynamic structure video.

Parameters	Video length [frames]	Pixel size [μm]	Field of view [$pixel^2$]	Blink density [%]	Blink density $\left[\frac{1}{\mu m^2}\right]$
Values	1000	0.16	32 x 32	0.2	0.78

Table S1: Filament simulation parameters.

1.2. Mitochondria simulation

Here we followed a similar scheme of simulation, but we changed the ground truth and the simulation parameters. First, we chosen N random center-of-mass (CM) positions for N mitochondria in the simulated field of view (FOV). Then, around each position we have chosen a random number of edge points from a uniform distribution of [30, 50] points. Each point was assigned with an angle in the range of [0, 2π] and a distance from the center of mass according to the known size of mitochondria (see Table S2). Finally, we acquired the ground truth structures of each mitochondrion by drawing a polygon based on the randomly chosen edge points.

Parameters	Video length [frames]	Pixel size [μm]	Field of view [$pixel^2$]	Distance from CM [μm]	Blink density [%]	Blink density $\left[\frac{1}{\mu m^2}\right]$
Values	1000	0.16	32 x 32	0.5 – 1.2	0.5	1.95

Table S2: Mitochondria simulation parameters.

We have simulated two types of dynamic movements for each mitochondrion: global shift, with velocities in the range of [-20, 20] nm per frame; and mitochondrion warping. The warping was done by choosing K edge points and move them periodically according to a sine function.

The blinking videos were simulated in a similar fashion to the simulations of filaments, but some parameters have changed (see Table S2).

2. Neural network architecture

Super spatio-temporal resolution reconstruction falls within the domain of sequence-to-sequence (seq2seq) objectives. In our case, the input is a sequence of high-precision localization maps of single molecules in an SMLM experiment, and the output is a sequence of images containing high-resolution reconstruction of the imaged structure.

Previous work has shown that combining information from multiple frames is beneficial in means of reconstruction accuracy and temporal resolution improvement[2,3]. However, the suggested algorithms are based on Convolutional Neural Networks (CNNs) which are sub-optimal solution for seq2seq objectives. A more commonly used architecture for seq2seq tasks is Recurrent neural network (RNN).

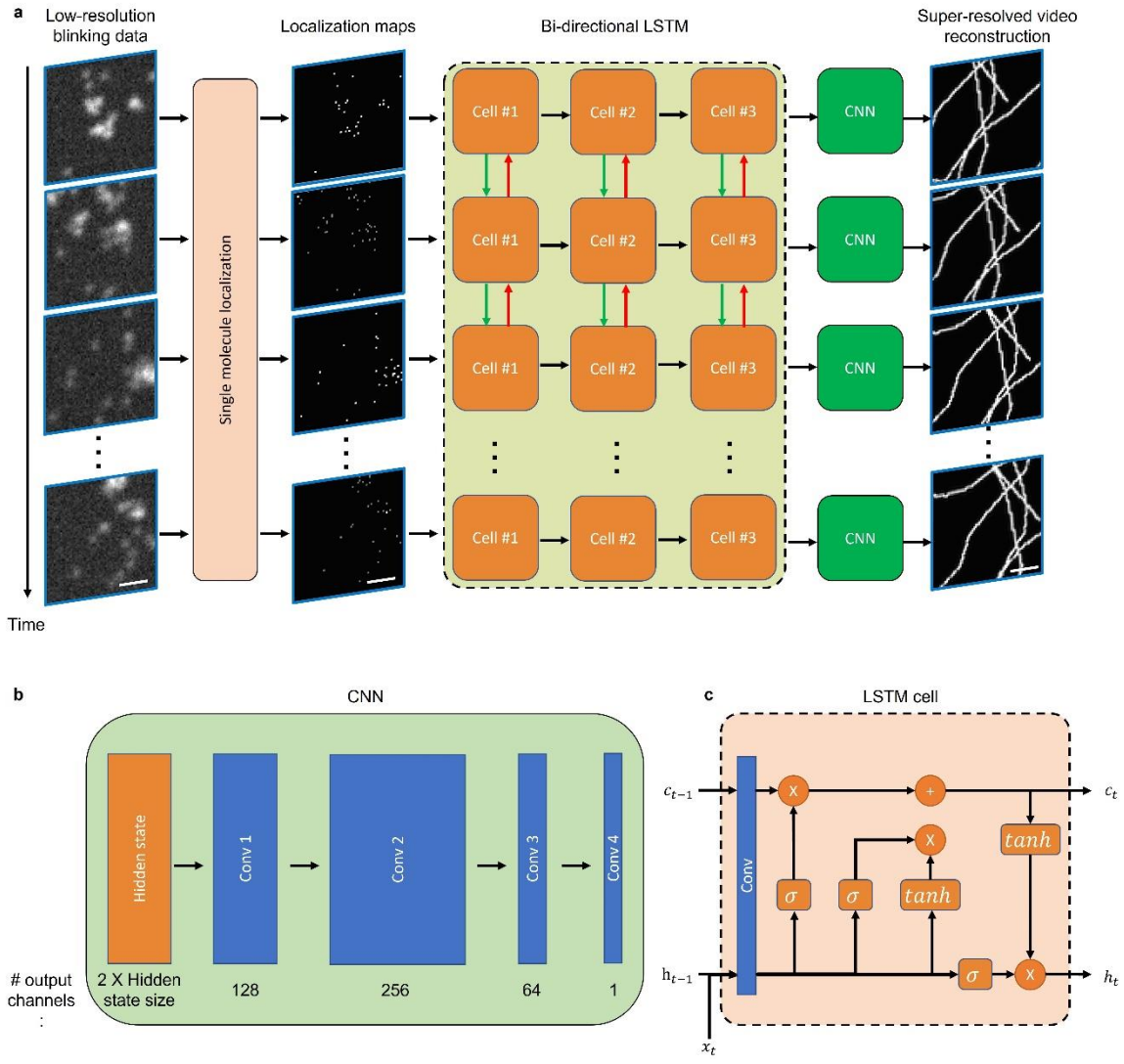

Figure S1: Neural network architecture. **a** We implemented bi-directional LSTM network with two layers. The first layer blocks get as input single low-resolution frames and the hidden states of the previous block. The second layer blocks output single super-resolved frames. In the forward pass (green arrows) the information propagates chronologically, while in the backward pass (red arrows) the images are inserted to the same network in reverse order. The output frames of both the forward and the backward pass are inserted to a CNN as two input channels. The output of the CNN is the super-resolved reconstruction of the entire video. **b** The architecture of the final CNN. **c** The architecture of the LSTM cells.

RNNs combine temporal information along the input sequence to provide better reconstructions in the output sequence. The weights in each RNN block are recycled during the inference process; therefore,

RNNs are composed of less parameters than CNNs. Nevertheless, RNNs outperforms CNNs in many seq2seq objectives. In our work, we implement a variant of RNNs named bi-directional long short term memory (LSTM) network (Supplementary Fig. S1). LSTMs are known for their ability to propagate important information throughout long input sequences. This advantage, along with the low memory demand, makes them perfect for the analysis of videos.

In addition to the suggested architecture, we have added another post-processing step to our analysis. In this step, we transform the output image to binary mask by defining all the pixels with values greater than some threshold as ones and the rest as zeros. Since the output image of the neural network $I(x,y)$ may be seen as a heatmap indicating the confidence of the network in the presence of a structure in each reconstructed pixel, we weighted each pixel in the binary map $B(x,y)$ according to the network confidence. Therefore, we drew a patch around each pixel and multiplied this patch by a 2D gaussian with standard deviation that equals to one over the original pixel value:

$$B(x_i,y_i) = \begin{matrix} 1\,, & I(x_i,y_i) > threshold \\ 0\,, & I(x_i,y_i) < threshold \end{matrix}$$

$$Final\ output(x,y) = B(x,y) \cdot \frac{1}{\sqrt{2\pi \cdot I(x_i,y_i)}} \cdot e^{\frac{-(x-x_i)^2-(y-y_i)^2}{2 \cdot I^2(x_i,y_i)}} \quad , x,y \in \{patch\ around\ x_i,y_i\}$$

This function would decrease the pixel intensity where the network confidence is low and maintain high pixel intensity otherwise. The resulting frame of this analysis would keep the intensity of high confidence pixels and reduce the intensity of low confidence pixels.

3. Accuracy quantification

3.1. Reconstruction accuracy quantification

The ground truth in our simulations were binary masks containing ones at simulated positions of emitters and zeros at the background. The outputs of our network were heatmaps containing different values in the range of [0, 1]. Higher pixel values meant that the network had higher confidence in estimating the structure at those pixels.

Finding the optimal accuracy measure for comparison between the predictions and ground truth is not a trivial task. The pixel-wise mean squared error (MSE) is a widely used measure for this purpose; however, in some cases it poorly describes the quality as we would expect. For example, when the sample is small relative to the FOV, most of the pixels in the ground truth image would have zero intensity. Therefore, consistently predicting matrices full of zero values would yield a very low error using the MSE. Structure similarity (SSIM)[4] will suffer from similar problems as MSE, due to the fact it relies on comparison between the mean intensity and standard deviation of the predicted image and the ground truth. Jaccard index[5] might be used to describe the similarity between two groups: the group of predicted localizations and the group of ground truth localizations. But in our case, we compare matrices and not localization lists and it is hard to compare between the predicted heatmaps provided by our neural network and the ground truth binary maps representing the sample.

Therefore, we decided to quantify DBlink performance on simulated data according to two different metrics: the reconstruction fidelity to the ground truth structure; and the network hallucinations displayed in its reconstructions. The reconstruction fidelity term is measured by the following steps: binarizing the predicted image based a predefined threshold of half the maximal intensity; counting the number of pixels marked as ones in both the predicted image and the ground truth; dividing that number by the total number of pixels marked as ones in the ground truth image.

(1) $$Fidelity\ [\%] = \frac{number\ of\ correctly\ classified\ structure\ pixels}{number\ of\ structure\ pixels\ in\ the\ ground\ truth} \cdot 100 = \frac{TP}{TP + FP}$$

Where P are the pixels containing structure in the ground truth, N are the pixels related to background in the ground truth, F means wrong classification (predicting structure as background and vice versa), and T means correct classification. The hallucination term was measured by the following steps: summing the number of pixels marked as ones in the predicted image and as zeros at the ground truth image (wrong structure predictions, namely FP); dividing this number by the number of pixels marked as zeros in the ground truth (correct background predictions and wrong structure predictions, TN and FP respectively)."(2) $Hallucination\ [\%] = \frac{number\ of\ wrongly\ classified\ structure\ pixels}{number\ of\ background\ pixels\ in\ the\ ground\ truth} \cdot 100 = \frac{FP}{TN+FP}$

In the experiment that contained unwanted stage drift, we quantified the accuracy as follows: First, we generated the ground truth image using Deep-STORM[3] reconstruction with drift correction and density filter tools. Next, we shifted back our reconstructed video frames according to the framewise drift prediction. Then, we binarized both our reconstructions and the ground truth reconstruction with thresholds that equal to the 75th percentile of each image intensity histogram (Supplementary Fig. S2). Finally, we quantified the reconstruction accuracy by measuring the cross-correlation between the re-shifted reconstructions $(\hat{y}_i)$ and the static ground truth image (y). The final normalized term we used is:

$$Accuracy_i = \frac{\max(\hat{y}_i \star y)}{\sqrt{(\hat{y}_i \star \hat{y}_i) \cdot (y \star y)}}$$

Where $\star$ marks the cross-correlation operator. The mean accuracy we obtained was 0.89.

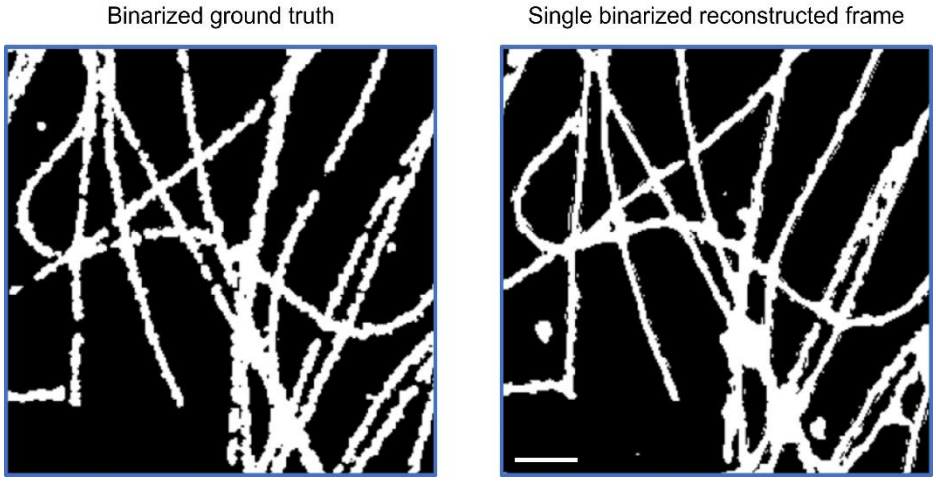

Figure S2: Quantifying the reconstruction accuracy in drifting sample experiment. Left: The ground truth structure obtained by Deep-STORM reconstruction in addition to application of drift correction and density filtration. Right: A single reconstructed frame of DBlink. Both images were binarized according to the 75th percentile of each image. Scale bar = 2 μm.

In addition to the similarity calculation between our reconstructed frames and Deep-STORM's reconstruction, we calculated the drift according to our reconstruction and compared it to the prediction we obtained using Deep-STORM drift correction mechanism (Supplementary Fig S3). We showed that both drift estimations agree with mean error of 38 [nm].

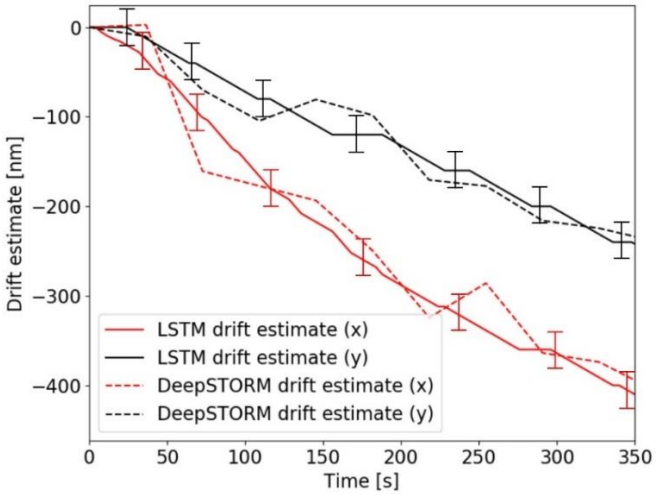

Figure S3: Deep-STORM drift prediction (dashed line) is consistent with DBlink's motion prediction (solid line); error bars, marked in black and red for x and y direction respectively, depict +/- standard deviation of the estimated drift over n=10 repetitive frames and a single frame (e.g. drift estimation between the first frame and frames 51-60).

In the experiment that contained camera rotation, due to the finite numerical limitation to exactly rotate and shift back each frame we quantified a different property of our reconstructions – the consistency. To do so, we have measured the cross-correlation between every two frames in the reconstructed video:

$$Accuracy_{ij} = \frac{\max\left(\hat{y}_i \star \hat{y}_j\right)}{\sqrt{\left(\hat{y}_i \star \hat{y}_i\right) \cdot \left(\hat{y}_j \star \hat{y}_j\right)}}$$

The result of this measurement is a matrix containing ones in the diagonal and normalized cross-correlations elsewhere. We achieved a mean consistency term of 0.91, over 20 neighboring frames, indicating that our reconstructed structure indeed maintains a high level of temporal uniformity throughout the reconstructed video (Supplementary Fig. S4).

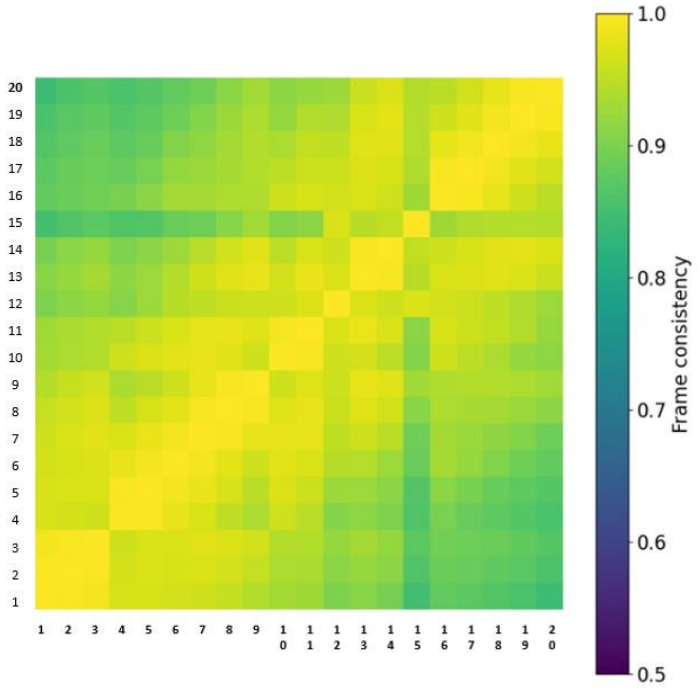

Figure S4: Consistency quantification. We measured the normalized cross-correlation between every two frames in the reconstructed video. The diagonal values mark the autocorrelations of each frame with itself; hence, they contain ones.

In the experiment containing dynein motors that moved on static microtubules, the hallucination quantification was similar to the experiment where unwanted drift has occurred. First we binarized ThunderSTORM and DBlink reconstructions according to the 75th percentile in each reconstructed frame. Then, we used eq. 1 to calculate the hallucination percentage in DBlink's reconstruction. We obtained mean hallucination percentage of 0.1 %.

3.2. Spatial resolution quantification

We quantified the spatial resolution according to Fourier ring correlation (FRC)[6]. In this method, we used DBlink reconstruction of static data along with super-resolution reconstruction of the same structure using Deep-STORM. Then, we multiplied the Fourier transform of each subsample. Finally, we measure the mean value of the multiplication image over rings with an increasing size. When the mean pixel intensity of a ring drops below a certain threshold, we mark the radius of that ring as the maximal spatial frequency that occurs in our reconstruction (Supplementary Fig. S5). The resolution is estimated by one over the maximal spatial frequency we achieved. We used the previously suggested 2σ threshold as our decision threshold. This threshold is computed by dividing 2 over the square root of half the number of pixels in each ring.

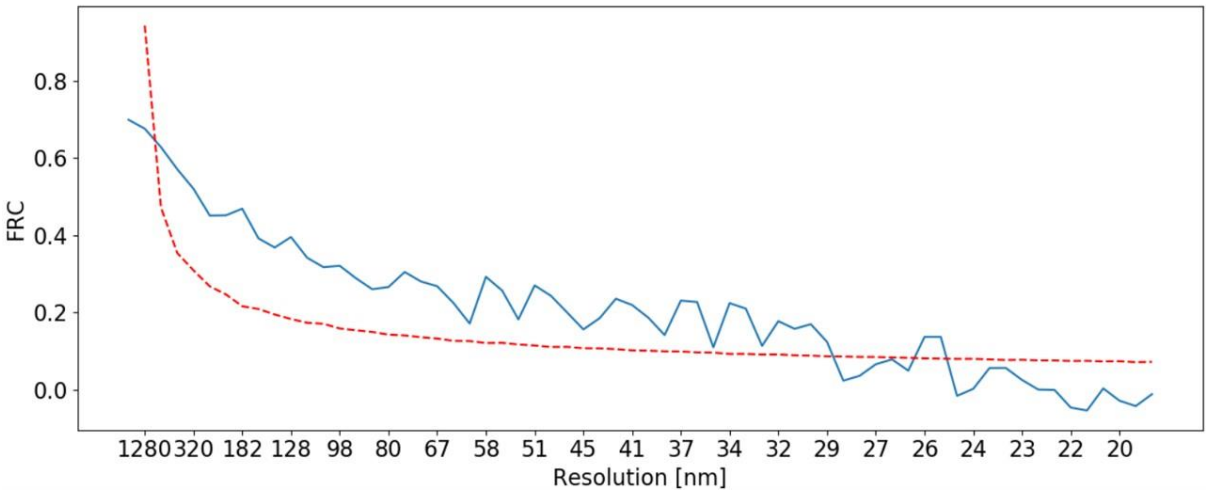

Figure S5: Fourier ring correlation analysis for spatial resolution quantification. First, we take two random subsamples of the data; then, we multiply the Fourier transform of the subsamples. Finally, we calculate the FRC according to the mean intensity value of all the pixels in a ring increasing in size. The resolution is determined according one over the cut-off frequency we achieved in the meeting point between the FRC curve and the predetermined threshold.

In addition to FRC, we also used decorrelation analysis[7] for resolution estimation. According to decorrelation analysis DBlink obtained spatial resolution of 30 nm (Supplementary Fig. S6).

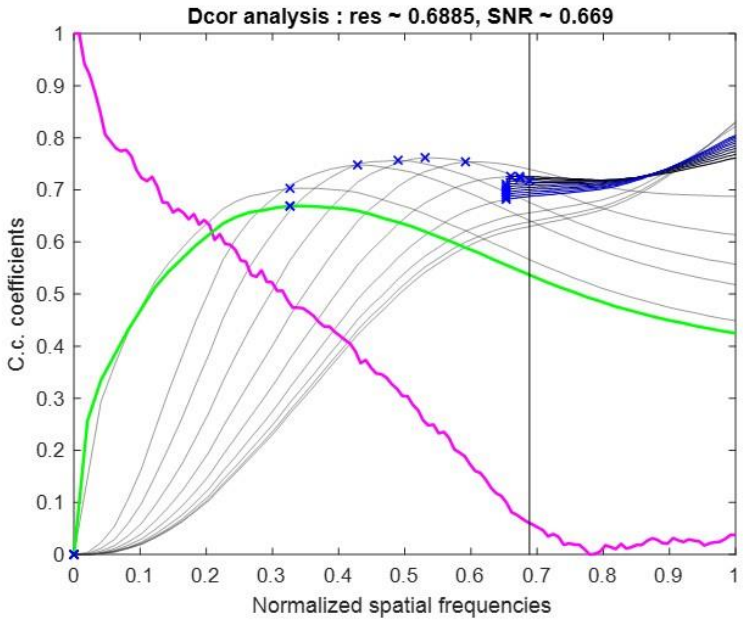

Figure S6: Decorrelation analysis. The output of decorrelation analysis algorithm published by A. Descloux et al[7]. The maximal spatial frequency in our reconstruction was the ~68th percentile of the maximal achievable frequency in our system. In our experiment this number matched spatial resolution of ~30 nm.

3.3. Additional performance quantification

In this section, we report additional simulations quantifying DBlink performance in case that the testing data deviates from the training data. Our first experiment was planned to rule out the existence of reconstructed features appearing at the wrong time, namely, appearance/ disappearance of ground truth features from past/ future frames. To test this, we generated new simulations containing appearing and disappearing structures (see Supplementary Video S16). We have quantified the temporal distance between the appearance/ disappearance of these structure in the reconstructed

video compared to their actual appearance/ disappearance time in the ground truth. The mean error of the appearance/ disappearance time between the ground truth and the reconstructed video is presented in Table S3. We use the notation “partial appearance” as rough quantification of cases in which some structure features occurred before the entire structure was visible; these cases are of interest since they indicate reconstruction artifacts from future frames. Analogously, we use the notation “partial disappearance” in cases a significant amount of structure features has disappeared from the reconstruction, but some features remain (reconstruction artifacts from past frames).

	Partial appearance error [frames]	Full appearance error [frames]	Partial disappearance error [frames]	Full Disappearance error [frames]
Mean error (signed)	-3.875	1.625	-2.875	2.5
Mean error (absolute)	3.875	3.875	2.875	3.25

Table S3: appearance/ disappearance error in simulated data. Negative sign indicates the appearance / disappearance of a structure before its appearance / disappearance in the ground truth data.

Table S3 shows that DBlink reconstructions present features from far past/ future frames within a temporal window of ~3 frames, which is equivalent, for example, to 45 ms in the case of the microtubule/ ER reconstruction presented in our paper. Finally, it is important to note that these simulations depict extreme cases, in which the entire structure appears/ disappears in a single frame. In the case of experimental data, where structures do not instantly appear/ disappear, more information would be available for DBlink to estimate the correct appearance/ disappearance time.

Additionally, we have tested DBlink robustness to more extreme deviations from the training data by simulations aiming to quantify DBlink performance when the amount of structural information per time point is reduced. To reduce the amount of structural information per time point, one can either make the structure move faster, or reduce the number of localizations per frame. First, we generated simulations of mitochondria moving in increasing velocities and checked the velocity effect on DBlink reconstruction fidelity to the ground truth (Supplementary Fig. S7).

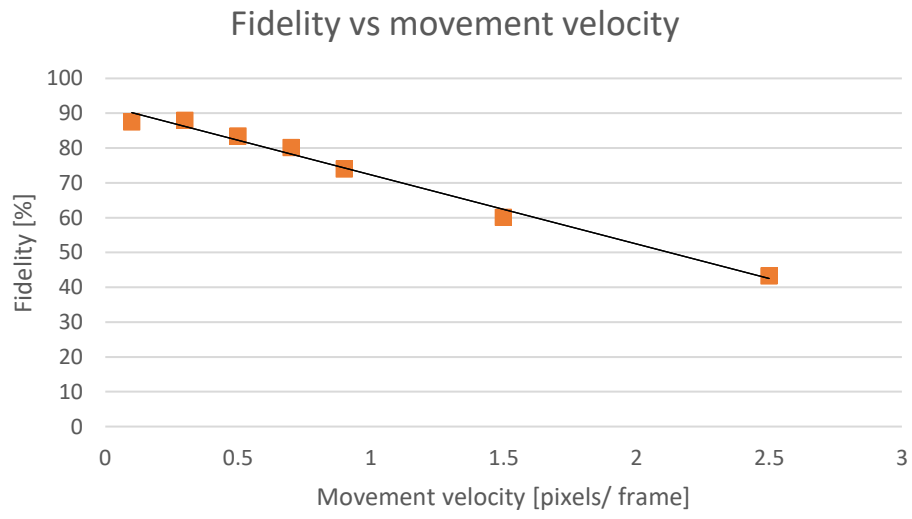

Figure S7: DBlink reconstruction fidelity to the ground truth vs mitochondrial movement velocity. The training data contained mitochondria-like structures moving slower than 0.5 pixels/ frame.

DBlink performance remained stable when the test data moved in similar velocities to the velocities presented in the training data (or slower). DBlink performance was significantly hindered (dropping from ~90% to 60%) when the movement velocity was 3 times higher than the maximal training velocity. Importantly, the reduced accuracy in DBlink reconstructions in this case, was not only due to deviation from the training data, but also due to the challenge in reconstructing very fast dynamics, where the blinking data is insufficient for sampling the motion correctly.

In addition to velocity variations, we have also tested the effect of varying the blinking density. Since the common definition of blinking density (number of emitters per area unit) is affected by the relation between the structure size and the field of view size, it sometime fails to describe the amount of structural information available per frame. For example, let's imagine two fields of view (FOV) of the same size; in one FOV there is a structure of size X and on the other FOV there is a structure of size $5 \cdot X$. Assuming the labeling density and the fluorophore blinking kinetics are the same for both structures, we would observe roughly 5 times more emitters in the second FOV than in the first FOV. Since in the FOV we have 5 times more emission emitters, the blinking density in the second FOV would be 5 times higher than in the first FOV. Namely, despite the fact that the blinking densities would be different, the mean percentage of structural information (which limits reconstruction algorithm performance) visible in each recorded frame would be the same (assuming that the fluorophores are homogeneously distributed along the structures). Therefore, we have used the following definition for the blinking density: the percentage of ground truth pixels that appears in each localization frame. For example: a 50% blinking density would mean that in each localization map, half of the pixels in the ground truth structure would be visible. In our simulations, we have varied the blinking density parameter and recorded the reconstruction fidelity to the ground truth (Supplementary Fig. S8).

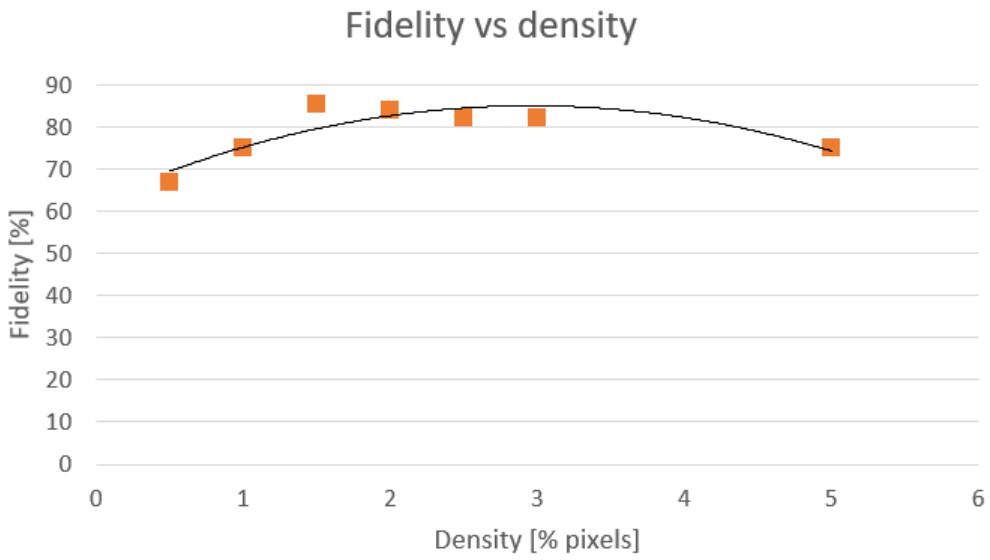

Figure S8: DBlink reconstruction fidelity to the ground truth vs blinking density DBlink was trained on blinking density of 2 % of the pixels in each localization map.

In this experiment, we have shown that deviations from the training blinking density have little effect on the reconstruction accuracy, namely, we observed ~10% reduction in the reconstruction accuracy when reducing the blinking density by a factor of 4 or by increasing the blinking density by a factor of 2.5 in compared to the training blinking density.

Finally, in order to test DBlink on biologically relevant dynamics at super-resolution, we have conducted live STED imaging of mitochondria, which served as the basis for a blinking simulation. In this experiment-based simulation, the ground truth structure was a STED reconstruction of mitochondrial dynamics, and we simulated the input localization maps for DBlink by randomly sampling this data. Reconstruction consisted of super-resolved dynamics at temporal resolution of 1.2 s per frame, corresponding to the STED frame rate, which is slow relative to the reported temporal resolution in our manuscript. Nevertheless, we are mainly interested here in the realistic structure and morphological changes that occurred in an experiment containing live-cell mitochondrial dynamics.

To generate the localization data from the STED ground truth video, we filtered noise artifacts using a median filter and binarized the image using a constant intensity threshold; then, we randomly chose structure pixels from the ground truth to serve as localizations per frame. The number of localizations per frame was chosen according to the blinking density we have used in our training data. Because of the low temporal resolution of the STED video, simulating a single localization frame per STED frame did not yield enough information to achieve a good reconstruction of the dynamics; therefore, we have generated 5 localization frames per STED frame. Next, we input the localizations to DBlink and compared its reconstruction to the ground truth STED video (Supplementary Fig. S9 and Supplementary Video S17).

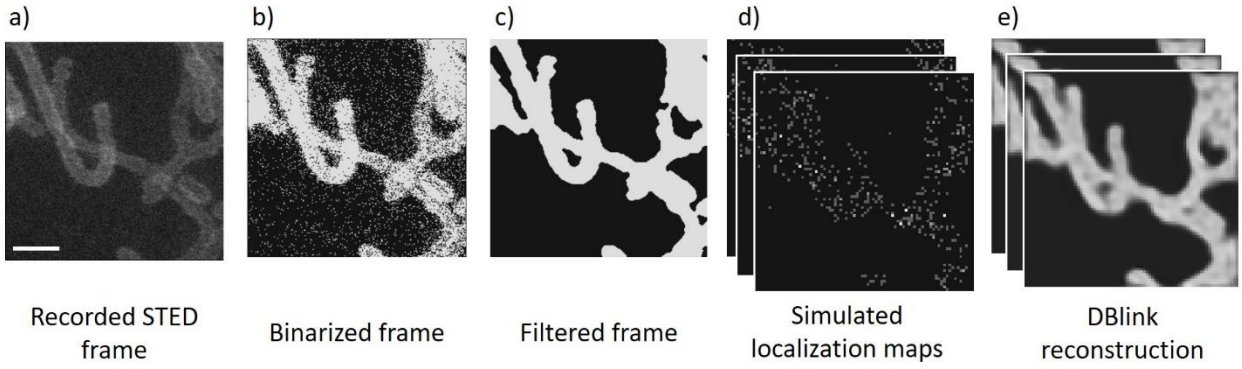

Figure S9: DBlink reconstruction of quasi-simulated data based on STED imaging of mitochondrial dynamics. a) recorded frame using STED; b) binarized data; c) ground truth structure after noise filtration; d) generation of multiple blinking frames; e) DBlink reconstruction based on the simulated blinking frames. Scale bar = $1\mu m$.

DBlink was able to reconstruct the mitochondrial dynamics in the STED ground truth data with mean fidelity score of 89.1% and mean hallucination percentage of 5.6%. Remarkably, no retraining of the network was needed, namely, the network could reconstruct morphologies and dynamics that never occurred in the training data.

To summarize this section, although DBlink is not error free, the experiments described above supply additional validation to the generalizability of DBlink to various changes in morphology and dynamics that did not occur in the training data. Furthermore, it is always possible to observe the confidence map outputs that DBlink provides for further inspection of reconstruction quality.

4. STED experiment sample preparation and imaging method

U2OS-TOM20-dHT7-CalR-HT7-KDEL stable cell lines (a kind gift from Julian Kompa and Kai Johnsson, MPI for Medical Research, Heidelberg) were cultured in T-74 flasks (Greiner, Germany) at 37°C and in 5% CO2 in Dulbeco's Modified Eagle Medium (DMEM)/F-12 (Gibco, Thermo Fisher, USA) supplemented with 10% (v/v) fetal bovine serum (FBS) (Coring, USA), 1% penicillin-streptomycin (w/v) (Gibco, Thermo

Fisher, USA), and 1% GlutaMAX (v/v) (Gibco, USA). 1-2 days before imaging, 2x104 cells were seeded on fibronectin-coated (Sigma-Aldrich, Germany) 8-well chambers (Sarstedt, Germany).

For STED live-cell imaging, cells were washed with pre-warmed 1x Dulbeco's phosphate-buffered saline twice. Then, 300 nM of Hy4-SiR dissolved in Live Cell Imaging Solution (LCIS, Thermo) was added to cells for imaging. STED imaging was performed on the Abberior STED Expert Line microscope (Abberior Instruments, Göttingen, Germany) equipped with an Olympus IX83 inverted microscope (Olympus, Japan) with an UPLXAPO 60x (NA 1.42) oil immersion objective (Olympus, Japan). Imspector software (v16.3.15507, Abberior Instruments, Göttingen, Germany) was used for microscope control and image acquisition. Fluorophores were excited by a 640 nm laser and the fluorescence depleted with a 775 nm pulsed laser. The fluorescence signal was detected in the wavelength range of 650 - 755 nm using avalanche photodiodes (APDs) and a gating of 0.75-8 ns. The pinhole was set to 1.0 AU and the pixel size was set to 20 nm. Images were recorded in line sequential mode with line accumulation of 1, and dwell time of 5 μs.

5. Window size optimization

We set the number of input frames for Deep-STORM windows and blind inpainting to be 300 frames. This number was chosen by optimizing the window size for the best reconstruction result (see Supplementary Fig. S10). The number of input frames for DECODE was similarly optimized to be 500 and 300 for the ER and microtubule experiments, respectively. The window size of eSRRF was chosen to be 250 frames and 100 frames for the ER experiment and the microtubules experiment respectively; eSRRF window size was optimized using the parameter sweep mechanism provided by the eSRRF ImageJ plugin.

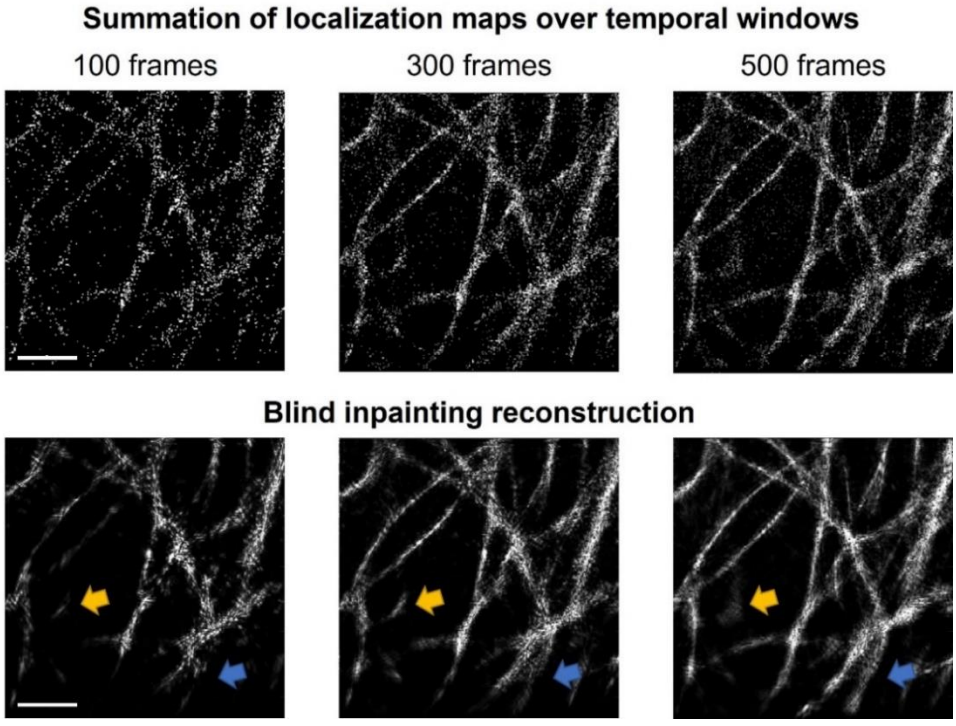

Figure S10: Blind inpainting evaluation[8]. Upper row: summation frame of localization maps over temporal windows with varying length. Bottom row: reconstruction of the summation frame by applying blind inpainting algorithm on it. All the reconstructions in the bottom row managed to filter noise and emphasize relevant features of the sample. However, summing 100 localization frames is not sufficient for the reconstruction of the entire sample; on the other hand, summing 500 frames generated motion blur that blind inpainting could not resolve

(yellow and blue arrows). Empirically, the best compromise between motion blur and reconstruction accuracy was obtained when we summed 300 frames. Scale bar = 2.5 μm.

6. Confidence hallucination calculation

Many neural-net-based reconstruction algorithms are usually considered as a black-box, and the outputs of these algorithms are hard to interpret. Hence, in this work we added an explainable output to DBlink acting as a guide in the interpretation of DBlink reconstructions (Supplementary Video S11, Supplementary Video S12, Supplementary Video S13).

The confidence measure we utilize is based on the pixelwise network predictions. We hypothesized that predicted pixel values that are farther than 0/255 (background/ structure values in the ground truth data) correspond to low network prediction confidence. To prove this claim, we have calculated the mean hallucination percentage in low confidence areas and in high confidence areas. We have defined high confidence pixel values as pixel values in the range of [0, 0.1] or in the range [0.9, 1], the rest of the pixel values are regarded as low confidence areas. The mean hallucination percentage was 1.5% in low confidence areas and 0.1% in high confidence areas. This indicates that the confidence measure we use correlates with prediction accuracy.

7. One-directional LSTM analysis

Before using bi-directional LSTM architecture, we first tested a one-directional LSTM architecture (Supplementary Video S14). As expected, the first few frames generated by the one-directional network did not contain enough structural information due to lack of accumulated information over the input sequence. Moreover, the ability of the one-directional network to detect and filter noisy localizations is lacking in comparison to the bi-directional network. It could be explained by the abundance of information from both the past and the future in the bi-directional network.

8. Supplementary video captions

Supplementary Video S1. Super spatiotemporal resolution reconstruction of simulated filaments. The temporal resolution of the reconstruction is 1 reconstructed frame per 10 simulated blinking frames.

Supplementary Video S2. DBlink reconstruction of a static STORM experiment exhibiting unwanted drift. Temporal resolution = 5 s (0.2 fps).

Supplementary Video S3. DBlink reconstruction of a static STORM experiment exhibiting global motion due to camera rotation. Temporal resolution = 0.8 s (1.25 fps).

Supplementary Video S4. Raw data of dynein motors (white) moving on static microtubules (red).

Supplementary Video S5. DBlink reconstruction based on Deep-STORM localizations of dynein motors moving on static microtubules reconstruction generated by ThunderSTORM[9]. Temporal resolution = 50 s (0.02 fps).

Supplementary Video S6. At the beginning of the video, we show Deep-STORM reconstructions of live-cell microtubules when summing localizations over temporal windows of lengths 500, 100, 20 frames. Finally, we show DBlink reconstruction at super spatiotemporal resolution. Spatial resolution = 30 nm; temporal resolution = 15 ms (66.6 fps).

Supplementary Video S7. Reconstruction of live-cell endoplasmic reticulum (ER). Comparison between DBlink and DECODE[10]. DBlink spatial resolution = 30 nm; temporal resolution = 15 ms (66.6 fps).

Supplementary Video S8. Reconstruction of live-cell microtubules. Comparison between DBlink at 66.6 fps and eSRRF[11] at 0.66 and 3.33 fps.

Supplementary Video S9. DBlink reconstruction of live-cell mitochondria over extended experiment duration of 12.5 minutes at super spatiotemporal resolution. Spatial resolution = 75 nm; temporal resolution = 500 ms (2 fps).

Supplementary Video S10. Focus on two regions of interest from the live-cell mitochondria sample. Spatial resolution = 75 nm; temporal resolution = 50 ms (20 fps).

Supplementary Video S11. Comparison between mitochondria training data and DBlink reconstruction of experimental data. The reconstruction contains new structures and motions never seen in the training stage, demonstrating the generalizability of the DBlink model.

Supplementary Video S12. Confidence map of reconstructed data in simulation.

Supplementary Video S13. Confidence map of reconstructed live-cell microtubule experiment. Spatial resolution = 30 nm; temporal resolution = 15 ms (66.6 fps).

Supplementary Video S14. Confidence map of reconstructed live-cell mitochondria experiment. Spatial resolution = 75 nm; temporal resolution = 500 ms (2 fps).

Supplementary Video S15. Performance comparison between one-directional and bi-directional LSTM.

Supplementary Video S16. Simulated videos of filament data. At a certain time point the experiment some filaments appear, and after some time they disappear. Left: localization maps (input to DBlink); Center: DBlink reconstruction; Right: Ground truth.

Supplementary Video S17. DBlink reconstruction of simulated data based on STED imaging of mitochondrial dynamics. Left to right: input localization maps; DBlink reconstruction; ground truth data generated by STED experiment; simulated diffraction limited data. Scale bar = 2 μm.

9. References

1. Shariff, A., Murphy, R. F. & Rohde, G. K. A generative model of microtubule distributions, and indirect estimation of its parameters from fluorescence microscopy images. *Cytometry Part A* **77**, 457–466 (2010).
2. Ouyang, W., Aristov, A., Lelek, M., Hao, X. & Zimmer, C. Deep learning massively accelerates super-resolution localization microscopy. *Nat Biotechnol* **36**, 460–468 (2018).
3. Nehme, E., Weiss, L. E., Michaeli, T. & Shechtman, Y. Deep-STORM: super-resolution single-molecule microscopy by deep learning. *Optica* (2018).
4. Wang, Z., Bovik, A. C., Sheikh, H. R. & Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* **13**, 600–612 (2004).
5. Jaccard, P. THE DISTRIBUTION OF THE FLORA IN THE ALPINE ZONE.1. *New Phytologist* **11**, 37–50 (1912).
6. Banterle, N., Bui, K. H., Lemke, E. A. & Beck, M. Fourier ring correlation as a resolution criterion for super-resolution microscopy. *J Struct Biol* **183**, 363–367 (2013).
7. Descloux, A., Grußmayer, K. S. & Radenovic, A. Parameter-free image resolution estimation based on decorrelation analysis. *Nat Methods* **16**, 918–924 (2019).

8. Wang, Y. *et al.* Blind sparse inpainting reveals cytoskeletal filaments with sub-Nyquist localization. *Optica* **4**, 1277–1284 (2017).

9. 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).

10. Speiser, A. *et al.* Deep learning enables fast and dense single-molecule localization with high accuracy. *Nat Methods* **18**, 1082–1090 (2021).

11. Laine, R. F. *et al.* High-fidelity 3D live-cell nanoscopy through data-driven enhanced super-resolution radial fluctuation. *bioRxiv* 2022.04.07.487490 (2022).