communications biology

A Nature Portfolio journal

Article

https://doi.org/10.1038/s42003-026-09899-y

DeepTRACE brings flexible machine learning to single-molecule track analysis

Oliver J. Pambos [1,2] ✉, **Jacob A. R. Wright** [1,2] & **Achillefs N. Kapanidis** [1,2]

Single-molecule imaging was developed to resolve behaviours obscured by ensemble averaging, but early tracking experiments typically captured only brief temporal windows, restricting analysis to individual states rather than the progression between them. Observation times now extend to minutes, revealing complete multi-stage biological processes that require new analytical approaches to capture sequences of events. Here we present DeepTRACE, a flexible tool for analysing single-molecule tracks in living cells that learns sequences of molecular events using past and future context from subcellular location, mobility, and photometric properties. It learns any molecular behaviour that can be annotated with natural-language labels, enabling users to tailor models themselves to specific biological questions without ML expertise. DeepTRACE generalises rapidly from very small datasets, training in minutes on a few hundred tracks, and supports extensive downstream analysis, including discovery of relationships absent from the training data. As DeepTRACE natively handles any numerical feature outside of its standard feature set, it incorporates photometric readouts, including measurements of internal conformation that reflect molecular action, alongside motion, temporal context, and subcellular location. We anticipate that researchers will use DeepTRACE to define biological states by molecular behaviour rather than mobility alone in complex multi-stage processes.

Single-Molecule Localisation Microscopy (SMLM) has provided stunning insight into biological processes, with single-molecule tracking among its most powerful modes[1]. Tracking experiments capture motion by connecting the sequence of a molecule's positions into a track. Such experiments have revealed fundamental processes, including chromosome target search[2] and Z-ring organisation[3].

Early tracking experiments were limited to the millisecond-to-second timescale due to fast photobleaching of the available fluorescent proteins; established analysis pipelines were thus designed for short, non-transitioning tracks, processing snapshots rather than full dynamics.

Recently, single-molecule tracks have been extended to multi-minute timescales, enabled by brighter and more stable probes[4,5], new labelling strategies[6,7], replenishing probes[8,9], and temporally-patterned excitation (stroboscopy, timelapse imaging). Such long-lived tracks promise to capture entire biological processes (e.g. transcription, cell division) unfolding over minutes, catalysing discoveries.

While traditional[10–16] and machine learning approaches[17–22] have advanced single-molecule track analysis, this potential has remained unrealised, largely because existing methods involve rigid constraints: pure diffusive states[10,13,17], truncation[11], memoryless transitions[12,15], fixed thresholds[11,16], fixed numbers[14] of rigidly-defined theoretical diffusion classes[17–21], and a lack of spatial context[11–22]. Biological processes, however, follow complex sequences of events, with actions depending upon past interactions, molecular conformation, and the cellular location at which they occur.

To address these limitations, we developed DeepTRACE (Deep learning-based Track Recognition, Analysis, and Classification of Events), a flexible, open-source software for track segmentation and analysis. DeepTRACE combines temporal and spatial context with photometric information and user-supplied natural language labels to train bespoke machine learning (ML) models, without coding or ML experience.

As a supervised method, DeepTRACE requires labelled examples to define functional states, either using simulated molecular tracks or direct annotation of real experimental data through DeepTRACE's interface which presents the video of each molecule alongside its 2D track and a synchronised feature time series (e.g. step size vs time). While manual annotation inevitably reflects human judgement and can be more difficult for unexpected or previously unseen biological behaviours, using experimental perturbations (e.g. antibiotics, protein over-expression) to enhance rare or hard-to-identify states both improves the reliability of human-generated labels and increases the density of key states. This enables more

---

[1]Clarendon Laboratory, Department of Physics, University of Oxford, Parks Road, Oxford, UK. [2]Kavli Institute for Nanoscience Discovery, University of Oxford, Sherrington Road, Oxford, UK. ✉e-mail: oliver.pambos@physics.ox.ac.uk

effective evaluation in the unperturbed system; a strategy common in Gamma Mixture Model (γMM) analysis[11].

Here, we demonstrate that DeepTRACE accurately segments long single-molecule tracks that capture multi-stage biological processes in both simulated and real experimental data. In both cases, DeepTRACE outperformed established methods for state segmentation and kinetic parameter inference. DeepTRACE generalises from small training datasets, often requiring only a few hundred tracks in a single field of view, and can be trained in minutes without specialised ML hardware or experience using either simulated ground truth or direct human annotation. The findings show that DeepTRACE provides a practical and flexible approach for analysing long single-molecule tracks, supported by a graphical interface and tools for quantitative analysis that assist biological interpretation of the resulting classifications.

Results and discussion

Overview of the DeepTRACE workflow

DeepTRACE combines tracking data with segmented cell boundaries to perform feature engineering; wherein powerful features (e.g. path straightness, distance to cell membrane) encoding higher order spatio-temporal or photometric properties, are constructed from raw features such as particle coordinates (steps 1–2, Fig. 1a). Tracks are annotated with natural language labels (step 3, Fig. 1a) through manual annotation or simulations. The user then selects features likely to reflect the biological process studied, and these train a custom ML model (steps 4–5, Fig. 1a). DeepTRACE segments unseen tracks using sequence-to-sequence classification with sliding window voting (Fig. 1b, Methods) enabling robust segmentation of tracks of any length. As feature selection is crucial, DeepTRACE's GUI offers extensive visual

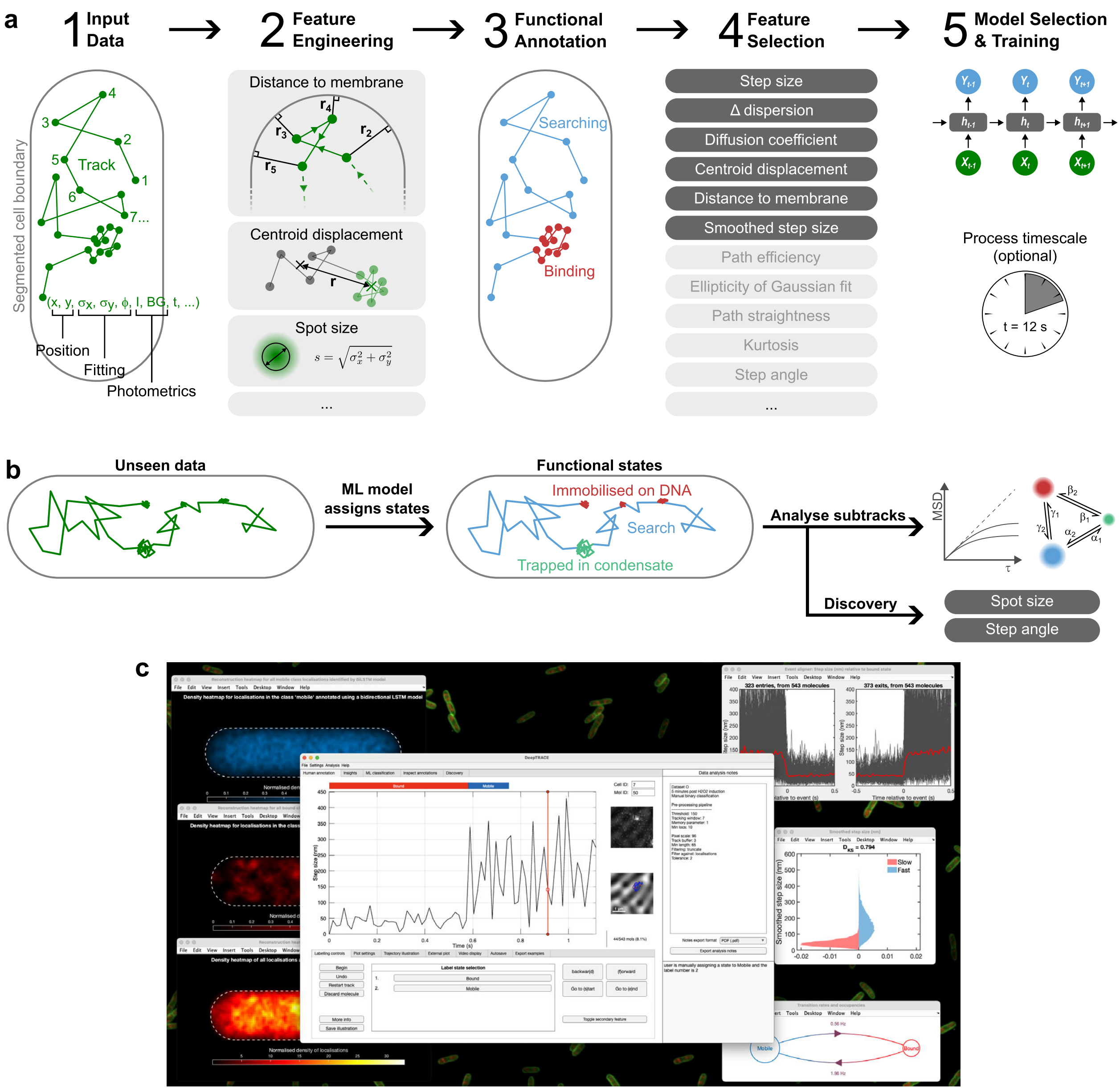

Fig. 1 | The DeepTRACE workflow. **a** Raw tracking, photometric, and cell segmentation data from a small test set are used to engineer a large number of features encoding different aspects of each molecule's behaviour. Biological states are annotated, either through human annotation of real experimental data or the ground truth from theoretical simulations. The user then identifies features likely to be predictive of the biological process, and these features are used to train a bespoke machine learning model. **b** The trained model segments unseen tracks into subtracks representing biological states, which are used in downstream analysis. Finally, the system identifies any unseen relationships between the data and features not used by the model, for hypothesis-free discovery. **c** DeepTRACE's graphical user interface displaying the Human Annotation tool performing binary classification of single molecule tracking data.

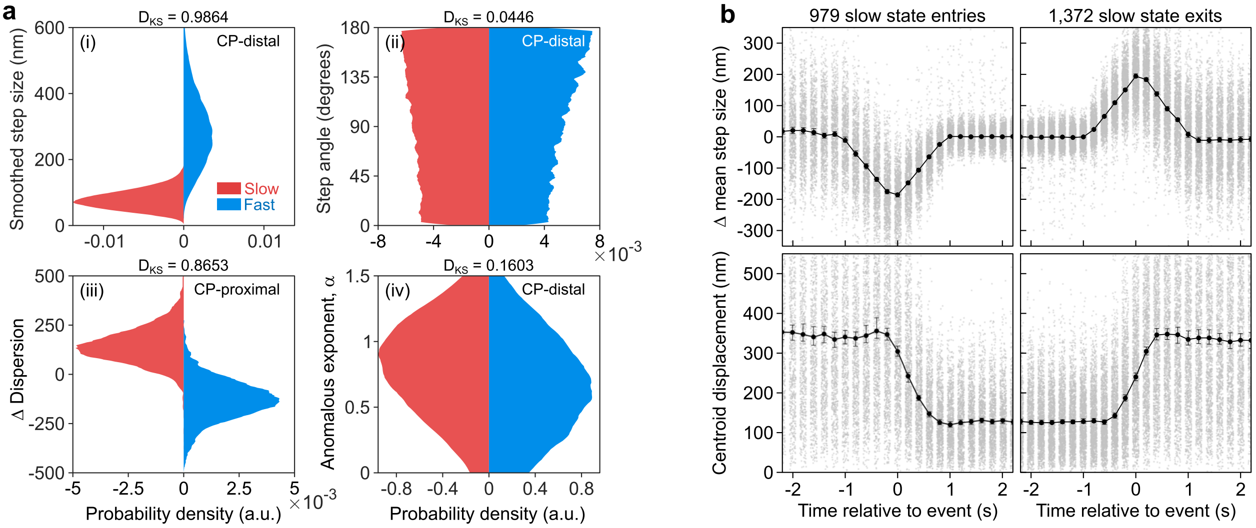

Fig. 2 | DeepTRACE visualisation of feature distributions. **a** Class-conditional feature distribution plots for four features, showing the Kolmogorov-Smirnov statistic between classes (metric above plots) acting as a proxy for the static discriminatory power of each feature, produced using DeepTRACE's Feature Visualiser tool. **b** Two features (Δ mean step size and centroid displacement) aligned relative to changepoints moving into ($n = 979$ independent events) and out of ($n = 1372$ independent events) the slow diffusive state using DeepTRACE's Event Aligner tool, revealing the temporal component of feature distributions. The line indicates the median, with upper and lower bounds at each timepoint representing the 95% nonparametric confidence interval of the median. Individual datapoints (grey circles) for each timepoint are shown for all event-aligned subtracks using uniform random jitter. Plots were produced using data from two-state reversible diffusion simulations (D = 0.01 $\mu m^2/s$ and 0.2 $\mu m^2/s$) containing $n = 1234$ independent tracks with ground truth class labels.

guidance to select powerful non-redundant features (Fig. 2, Supplementary Fig. 1).

Evaluation using simulated intracellular dynamics

We evaluated DeepTRACE using both simulated and real experimental data. To provide an unambiguous ground truth, we first simulated realistic particle motions inside a 3D model cell (Fig. 3a) using Smoldyn[23]. We modelled rod-shaped bacteria, where tight confinement masks the anomalous diffusion exponent that simplifies motion classification in larger eukaryotic systems (Supplementary Fig. 2). A series of fluorescence video frames was constructed from these 3D tracks using SMeagol[24], incorporating realistic experimental properties (e.g. PSFs, intra-frame transitions, motion blur, read noise). We processed the videos using a typical tracking pipeline[11,25], introducing sources of noise present in real particle-tracking applications, and imported the tracks and ground truth into DeepTRACE.

We simulated a diverse range of behaviours: reversible transition between diffusive states (Fig. 3b, Supplementary Fig. 3, Supplementary Movie 1), irreversible transport across the inner membrane (Fig. 4, Supplementary Fig. 4, Supplementary Movie 2), and reversible binding to a polarly-localised plasmid (Fig. 5, Supplementary Fig. 5, Supplementary Movie 3). Tracks were segmented by a DeepTRACE model incorporating a BiLSTM network with a self-attention mechanism, which learns the complex relationships between features through sequences of events. Segmentation of two-state diffusion data (step size time series, Fig. 3b, Supplementary Fig. 3) showed strong agreement with ground truth, with DeepTRACE consistently outperforming ResAnDi2, a top-ranked model in the 2nd Anomalous Diffusion Challenge[26].

After model segmentation, DeepTRACE derives global kinetic and spatial parameters (e.g. diffusion coefficients, occupancies, spatial profiles) by analysing the resulting subtracks. For example, subtracks are aggregated into class-specific MSD-lag time relationships from which apparent diffusion coefficients are estimated (see Methods). Similar results were obtained for inference of global kinetic parameters, compared to the established methods of γMM diffusion analysis[11], the widely-used software Spot-On[14], and ResAnDi2. DeepTRACE outperformed each method across all diffusion simulations (Figs. 3c and 6). This superior performance remained even with DeepTRACE restricted to small training datasets (160 ± 8 tracks), small window size (25 frames), and rapid training (5–10 min); all achieved without specialised ML hardware (laptop CPU; bars above plot Fig. 3c). By varying dataset size and the separation of mobilities we found that DeepTRACE is capable of accurate segmentation using datasets equivalent to single field-of-view recordings, and with diffusion coefficients of common DNA-binding proteins (Fig. 7). Higher performance (F1-score: 0.991, Jaccard: 0.982) occurred in irreversible transmembrane transport simulations (Fig. 4), likely reflecting the models' ability to learn sequences of events.

Learning and generalisation from human annotation

To assess DeepTRACE's ability to learn directly from human annotation, we generated two-state simulations using experimentally-established diffusion coefficients ($D_{repair} = 0.01\ \mu m^2/s$, $D_{search} = 0.64\ \mu m^2/s$) for DNA polymerase I (Pol1)[25]. 197 tracks were manually annotated using the built-in Human Annotation tool and used to train a DeepTRACE model (167 training, 30 validation tracks). The model was then used to segment 1292 tracks from a completely independent simulation, and DeepTRACE inferred kinetic parameters from the resulting subtracks. DeepTRACE returned values within 0.6–7% of ground truth, outperforming Spot-On, γMM and ResAnDi2 (Supplementary Table 1), and demonstrating accurate inference of realistic kinetic parameters when trained exclusively on human annotation.

We extended this approach to real experimental data, training a model on human annotation of 453 tracks from previously published experiments[25] of Pol1 repairing H_2O_2-induced DNA damage; a process involving ~2-second transient immobilisation events[16]. The model subsequently annotated unseen data from independent experiments, showing strong agreement with separate expert human annotations (Fig. 3d, Supplementary Fig. 6, Supplementary Movie 4). We also used the same model trained on human-annotation of real tracks to segment and analyse simulated data, for which an objective ground truth exists. Remarkably, despite differences between experiment and the synthetic training set (e.g. reversed class balance, different localisation error, free diffusion of the simulated slow state, different binding times) the model again outperformed analysis of the same dataset by established methods (Supplementary Table 1).

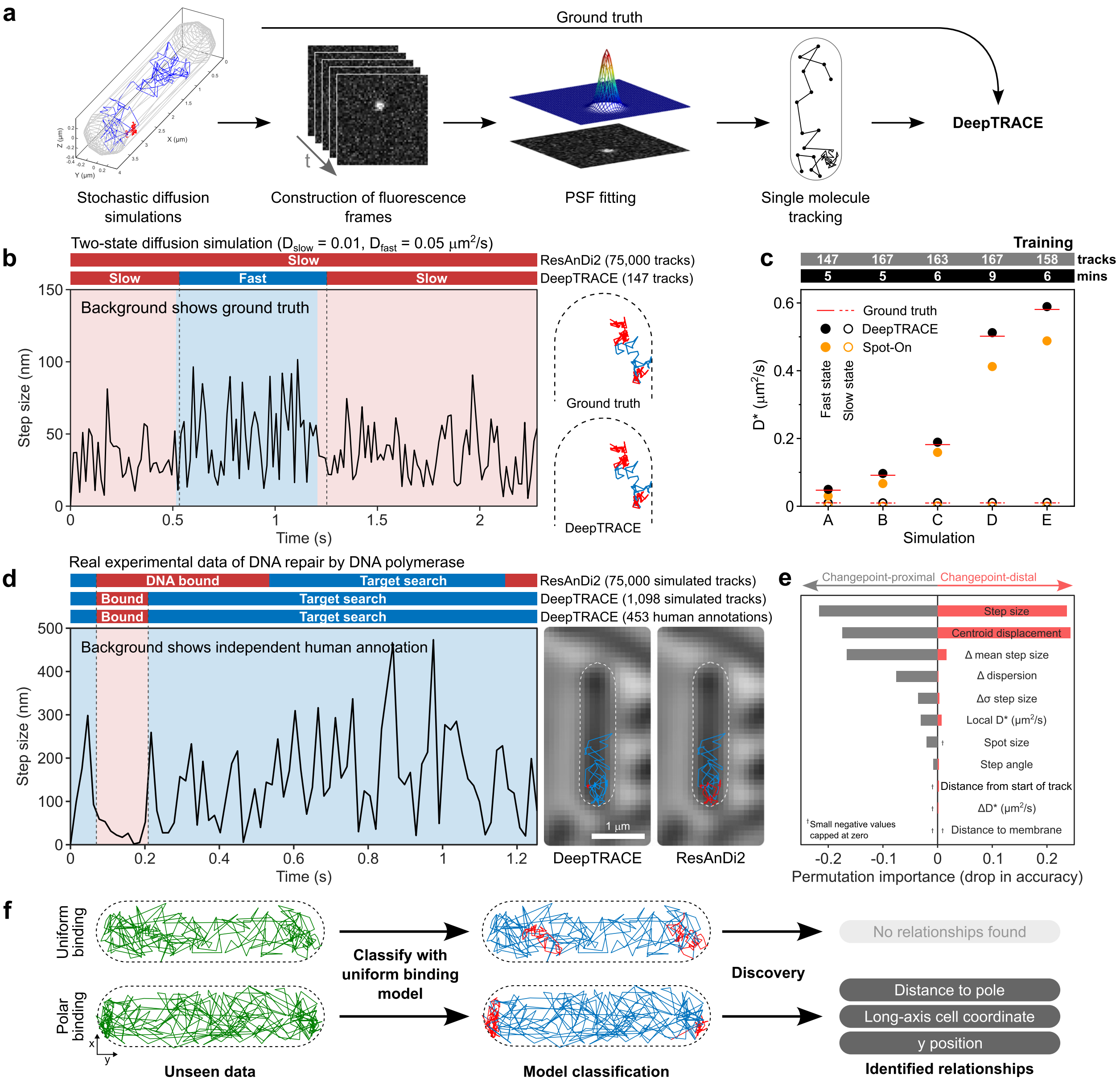

Fig. 3 | DeepTRACE analysis of simulated and real experimental data. **a** The simulation pipeline; particle simulations with Smoldyn were rendered into fluorescence video frames using SMeagol, and processed using a typical tracking pipeline. **b** Time series of simulated reversible switching between two diffusive states (D = 0.01 and 0.05 µm^2/s), showing classifications from a 4848-parameter DeepTRACE model trained on 147 tracks (bars above), ResAnDi2, and ground truth (background colours). **c** Performance comparison between DeepTRACE (black circles), and Spot-On[14] (orange circles) for inference of diffusion coefficients across five two-state diffusion simulations, compared to ground truth (red lines), using models trained with n = 147, 167, 163, 167, and 158 independent tracks for datasets A–E in 5, 5, 6, 9, and 6 min respectively on a single CPU core (bars above). **d** Time series of real Pol1 tracking in live *E. coli*[25] showing classifications from simulation- and human-trained DeepTRACE models and ResAnDi2 (bars above), compared to expert human annotation (background colours). Human-trained and simulation-trained models contained 2,172 and 4,577 parameters respectively. **e** Permutation importance showing the relative influence of features on classification of Pol1 tracks using a surrogate model, obtained from the mean drop in macro-averaged accuracy across n = 5 independent feature permutations. **f** DeepTRACE's Discovery tool evaluating n = 1249 simulated tracks of binding to a polar-localised plasmid using a model trained on uniformly distributed binding events from n = 1048 independent tracks; identifying the existence of the new relationship to pole proximity absent from training data. The raw fluorescence videos and track annotation sequences for the molecules shown in (**b**), (**d**), and the two examples in (**f**) are available in Supplementary Movies 1, 4, 5 and 6 respectively.

Discovery, efficiency, and practical considerations

To discover unexpected relationships, DeepTRACE's Discovery tool evaluates mutual information between classified states and features not used by the model, enabling hypothesis-free discovery. We illustrate this with simulations of a protein reversibly binding to a slowly diffusing plasmid located within the cell end-caps, transiently adopting the plasmid's mobility and spatial profile (Fig. 5). Using a model trained on binding events distributed uniformly across the intracellular space, the Discovery tool correctly identified a relationship between pole proximity and diffusive state despite this relationship being absent from the model's training data (Fig. 3f, Supplementary Movies 5 and 6). No such relationships were identified in control data where events were uniformly distributed.

Efficient learning from very small datasets arises mostly from the compression of key information into powerful engineered features (Fig. 7a inset), user-informed feature selection, and temporal resampling (Methods). Segmentation improves sampling in MSD calculations, removes track truncation and track-length biases, and better separates diffusive states (Supplementary Fig. 7,8). DeepTRACE performs diffusion calculations

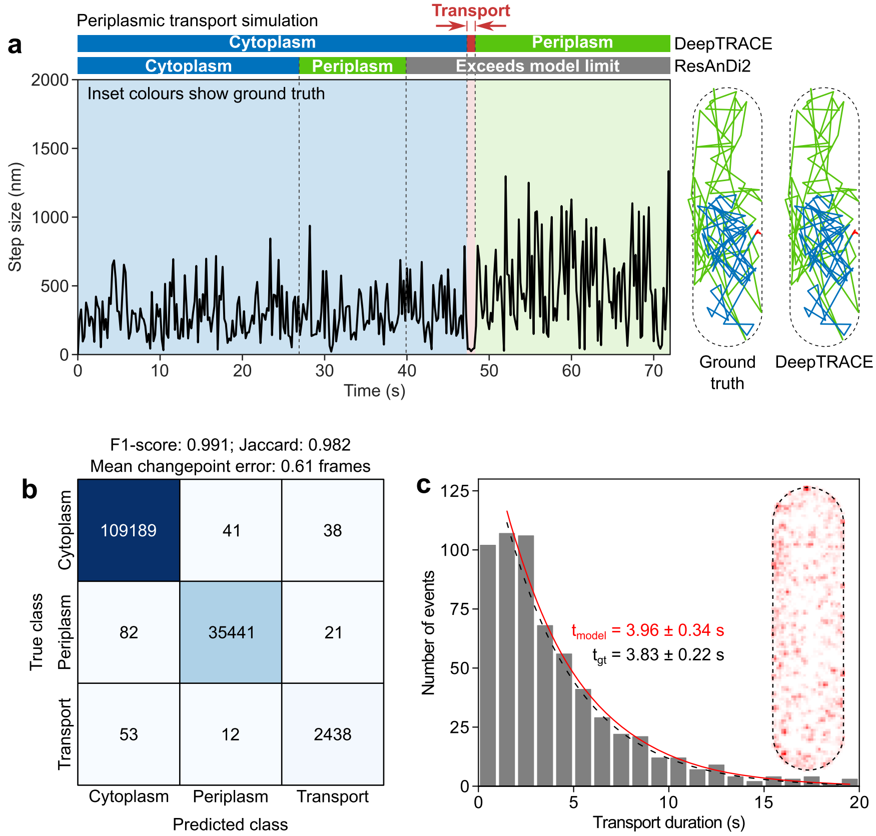

Fig. 4 | Analysis of simulated irreversible transport across the cytoplasmic-periplasmic membrane. **a** Time series of the step size feature for a simulated membrane transport event, showing classification from a 2273 parameter DeepTRACE model trained on $n = 1298$ independent tracks, and ResAnDi2 (bars above), compared to ground truth (inset colours). Shown to the right is the track plotted within the cell boundary (dashed line) coloured according to DeepTRACE and ground truth classifications of each timestep using DeepTRACE's Track Inspector tool. The time series has been cropped either side of the transport event to improve visual clarity; the full track is shown in Supplementary Movie 2. Supplementary Fig. 4 shows an expanded set of examples. **b** Confusion matrix showing classifications from the same DeepTRACE model of all localisations across $n = 306$ independent tracks in the evaluation dataset, with performance quantified by macro-averaged F1-score, Jaccard, and mean changepoint error (statistics above). **c** Residence times and spatial mapping (inset 2D map) of $n = 613$ model-identified transport events from $n = 1528$ independent tracks using DeepTRACE's State Analyser and Spatial Mapping tools, with segmentations obtained using a 2273-parameter DeepTRACE model trained on $n = 260$ independent tracks containing $n = 117$ examples of transitions into the transport state. Solid red and dashed black lines show single-exponential decay fits to model-identified and ground truth residence times respectively. All tracks used for training were obtained from a completely independent simulation to the evaluation data.

directly on subtracks, contrasting the standard approach of de-mixing populations from ensemble distributions, achieving higher accuracy with fewer tracks. Consequently, a single field of view is sufficient to infer global properties (Fig. 6, Supplementary Table 1) traditionally requiring ~80,000 tracks[25].

The ability to further enhance changepoint detection through experimental perturbations was demonstrated by increasing class transition rates in two-state diffusion simulations and evaluating on separate unperturbed data (Supplementary Fig. 9a). This perturbation, while leaving the underlying state properties such as diffusion coefficients unchanged, produced a model that achieved lower changepoint error on unperturbed data than an identical model trained on the same number of unperturbed tracks. This result demonstrates that DeepTRACE forms a more accurate internal representation of a changepoint from training on the perturbed dataset. Opportunities for such perturbations arise experimentally and can be a powerful practical approach to train models to recognise rare events; for example in Lagage et al.[25] where H_2O_2 exposure transiently increases the number of DNA damage and repair events without altering the underlying repair pathways.

We have restricted comparisons to diffusion analysis as existing pipelines are limited to diffusion tasks; however DeepTRACE can map any feature-class relationship, regardless of whether the target behaviour is spatial, mobility-related, photometric, or a combination of these properties. DeepTRACE also supports import of any arbitrary feature beyond its existing feature set (e.g. FRET efficiency which encodes molecular conformation for smFRET tracking, or orientation features from anisotropy readouts which capture rotational dynamics[27]) if provided as a separate column in the tracking file; ensuring adaptability to future applications.

DeepTRACE requires cell boundary information to construct spatial features and is currently optimised for rod-shaped bacteria using two-dimensional tracking data, with future plans to extend to 3D tracking modalities. Further developments include support for other cell geometries and an offline user-feedback system via the Track Inspector tool, allowing users to flag or correct misclassified track segments during visual inspection, with these annotations being incorporated into subsequent retraining rounds to improve performance on challenging behaviours. DeepTRACE's model interpretability tools could be further expanded using confidence estimation methods, such as those demonstrated by Seckler and Metzler[21].

In conclusion, DeepTRACE replaces rigid track analysis with a flexible, context-aware tool that interprets the complex sequences of events in long single-molecule tracks, extracting biological insights from minimal data, without specialised ML hardware or expertise. DeepTRACE's lack of hardcoded rules or pre-training, and its native handling of arbitrary features and classes, ensures future relevance as experimental methods evolve.

Methods

DNA polymerase I experimental data

Previously published single-molecule fluorescence tracking experimental data of DNA polymerase was kindly provided by Lagage et al.[25] in the form of TIF-formatted fluorescence video files of single-molecule tracking under continuous imaging with an exposure time of 15 ms, brightfield reference images, and MicrobeTracker-formatted cell segmentations. Fluorescence excitation of the organic dye JF549 covalently linked to DNA polymerase I via a HaloTag linker, was performed using variable angle epifluorescence excitation with a 561 nm laser. Under-labelling ensured temporally separated single-molecule tracks. In each case we used the first data acquisition following H_2O_2 exposure (4–7 mins) for model training.

Simulations

Simulations of single-molecule tracking experiments were generated using the Brownian dynamics simulator Smoldyn, with synthetic microscopy video sequences rendered using the software SMeagol.

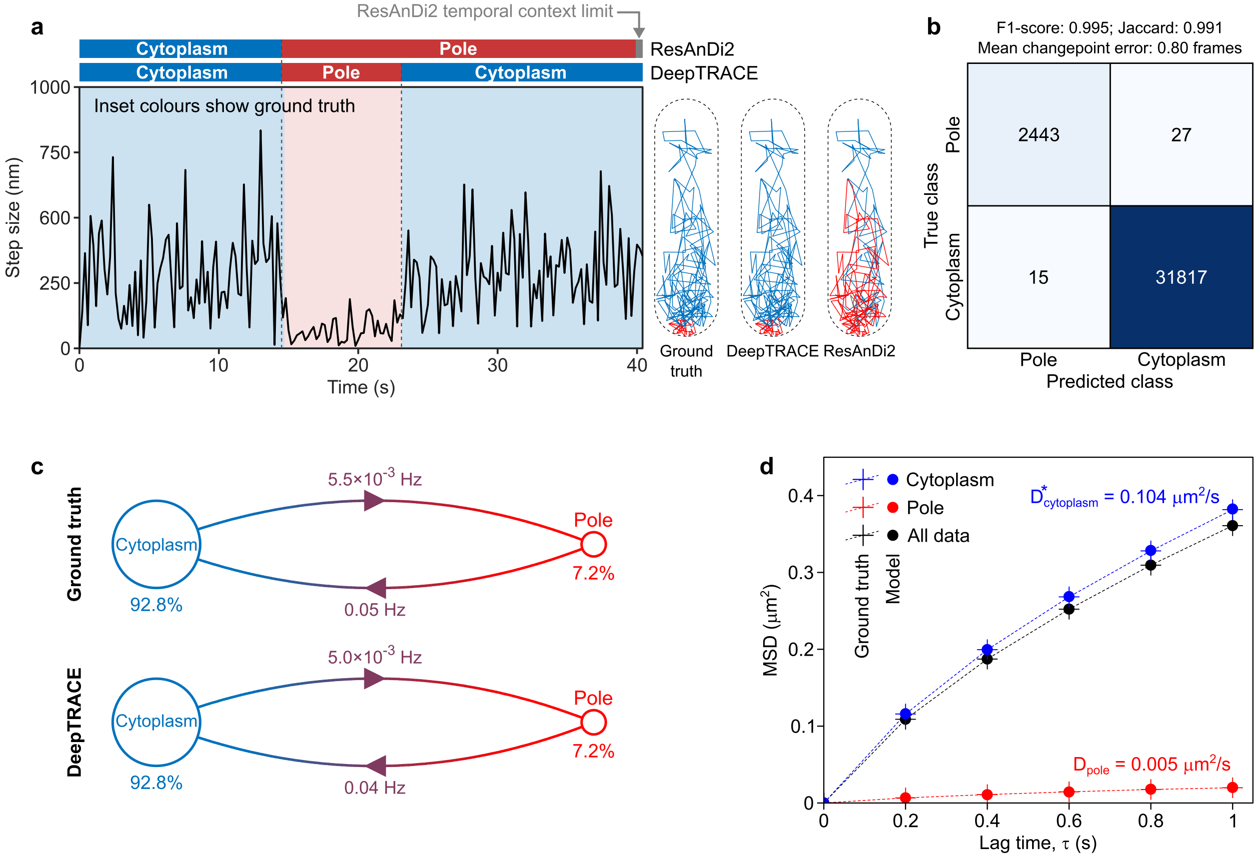

Fig. 5 | Analysis of simulations of reversible binding to a polar-localised plasmid. **a** Time series of the step size feature showing classifications using a simulation-trained 2252-parameter DeepTRACE model trained on n = 1061 independent tracks, ResAnDi2 (bars above), and ground truth (inset colours), produced using the Track Inspector tool. Shown to the right is the associated track plotted within the cell boundary (dashed line), coloured according to ground truth and model classifications of each timestep; the raw fluorescence video and track annotation sequence for this molecule is shown in Supplementary Movie 3. Supplementary Fig. 5 provides an expanded selection of examples. **b** Confusion matrix showing DeepTRACE classifications for all localisations across n = 209 independent tracks in the evaluation dataset, with performance quantified by F1-score, Jaccard, and mean changepoint error (statistics above). **c** Transition diagrams generated with the Transition Viewer tool for the same DeepTRACE model and ground truth segmentations, showing occupancies of each class (circle areas and percentages), and transition rates (frequencies adjacent to arrows). **d** MSD-lag time plots generated using DeepTRACE's Diffusion Analysis tool for DeepTRACE-segmented subtracks from n = 209 independent tracks (blue and red circles indicate the cytoplasmic and plasmid-bound polar states, respectively), unsegmented tracks (black circles), and ground truth (crosses connected by dotted lines). All tracks used for training were obtained from a completely independent simulation to the evaluation data.

Smoldyn simulations were performed inside an idealised rod-shaped cell, modelled as a cylinder (length 2.2 μm, radius 0.4 μm), and enclosed at both ends with hemispherical end caps of radius 0.4 μm (total length 3 μm). The surrounding simulation volume consisted of a regular cube of side 5 μm. 3D coordinates and diffusive states were computed at 1 ms and 10 ms intervals for inter-frame times of 15 ms and 200 ms respectively.

In both real and simulated data, changepoints were not synchronised to camera shutter actuations; consequently the vast majority of localisations immediately preceding changepoints comprise a mixture of states. This lack of synchronicity in both real and simulated data unavoidably contributes to the changepoint error, and we also note the existence of a class-asymmetry artefact that arises from the classification of different diffusion coefficients with sub-frame changepoints which further complicates classification. To assign a single ground truth class from the constituent subframes containing mixed classes we therefore explored several approaches, including the use of median, modal, and maximum diffusion values; settling on a ground truth definition as the state of the first subframe of each frame.

Fluorescence frames were generated by SMeagol from the coordinates and diffusive states exported by Smoldyn, using integration of fluorescence emission over 10 ms, at intervals of 15 ms to match common continuous imaging frame rates, and intervals of 200 ms for timelapse imaging. Simulated photoactivation ensured that only one molecule is active at any given time, and fluorescence emission continued until permanent photobleaching. Fluorescence intensities were sampled from a Gaussian distribution, with a simulated 2D Gaussian point-spread function (σ = 150 nm). Poisson-distributed background noise and camera simulation parameters (e.g. gain) were tuned to reproduce imaging data comparable to a typical super-resolution fluorescence microscope. Photobleaching times for each experiment were uniformly distributed across the range 65–150 frames. The photobleaching time for each molecule in the simulation was then drawn from a Gaussian distribution centred around this time.

Videos for each molecule were constructed in stacks of 500 frames. These videos were stitched together spaced by ten empty frames for continuous imaging and five empty frames for timelapse imaging, producing aggregated video sequences of 25,245 and 25,490 frames respectively. Datasets for all simulations were generated in independent batches of 300 and 2,000 molecules. To explore the variation of model performance with dataset size, training datasets were reduced by downsampling via random track elimination, and expanded by concatenation.

Three biological scenarios were simulated: (i) reversible interconversion between two diffusive states, (ii) binding to a polar-localised plasmid, and (iii) irreversible transport across the cytoplasmic-periplasmic membrane.

(i) Simulations of reversible interconversion between two diffusive states

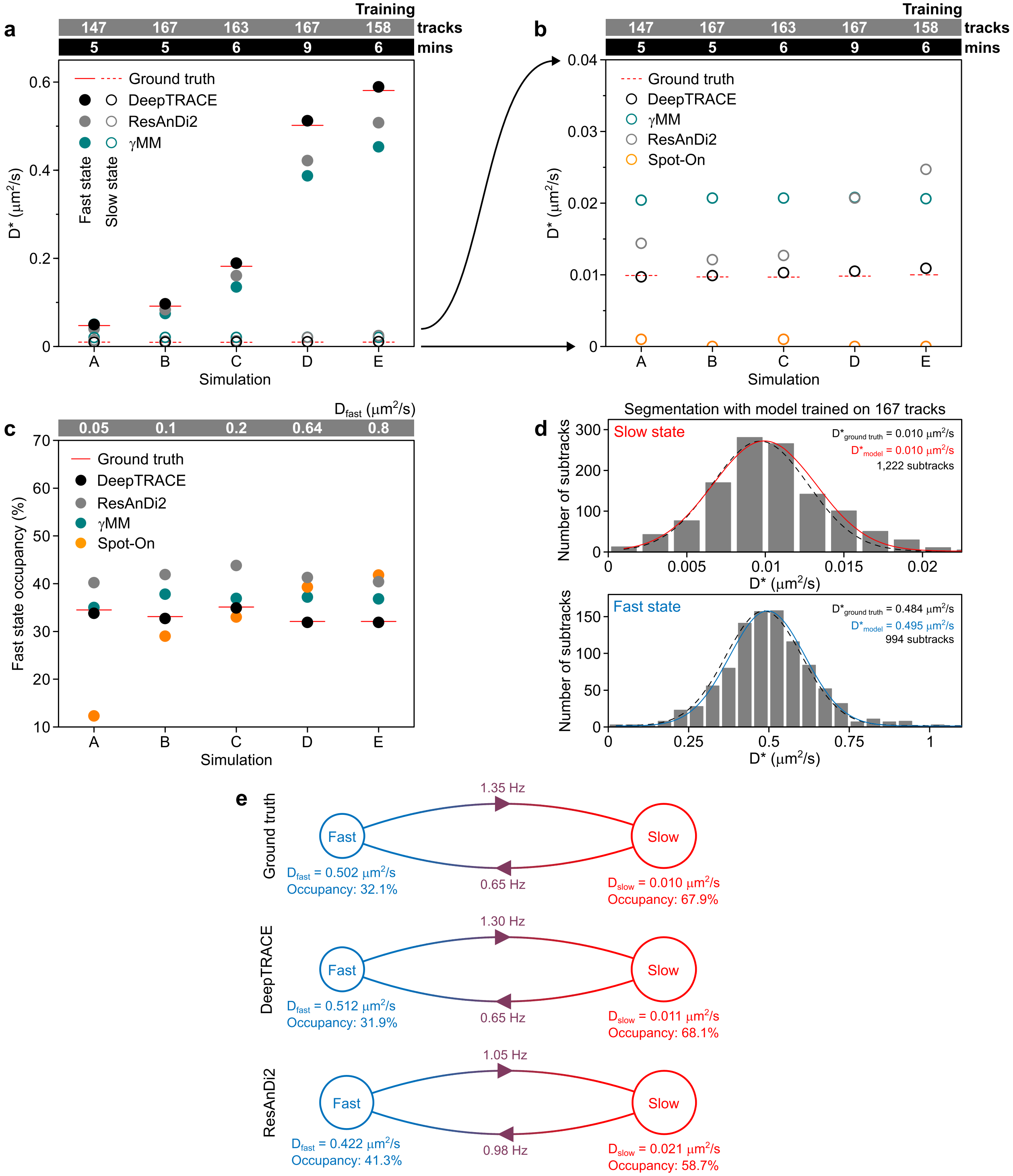

Fig. 6 | Performance comparison of DeepTRACE with established approaches. **a** Diffusion coefficient inference across five reversible two-state simulations by DeepTRACE (black circles), ResAnDi2 (grey circles), γMM (teal circles), and ground truth (red lines); results from Spot-On are shown in Fig. 3c (orange circles). DeepTRACE was trained on n = 147, 167, 163, 167, and 158 independent tracks for datasets A–E in 5, 5, 6, 9, and 6 min respectively on a single CPU core (bars above). **b** Enlarged view of slow-state inference. **c** State occupancies across five fast-state inputs $D_{fast} = 0.05, 0.1, 0.2, 0.64, 0.8$ μm²/s for datasets A-E respectively (bars above), reduced by confinement; with $D_{slow} = 0.01$ μm²/s. **d** Apparent diffusion coefficient histograms from the Diffusion Analysis tool (slow, top; fast, bottom), with Gaussian fits to model-classified (coloured lines) and ground truth subtracks (dashed black lines); the model was trained on $n = 167$ independent tracks. Histograms show $n = 1222$ (slow) and $n = 994$ (fast) segmented subtracks from dataset D. **e** Transition diagrams using DeepTRACE's Transition Viewer tool, showing occupancies (circle area) and transition rates (adjacent to arrows). Test datasets A–E contained $n = 1307, 1371, 1272, 1292$, and 1299 independent tracks respectively, and were generated completely independently from training datasets.

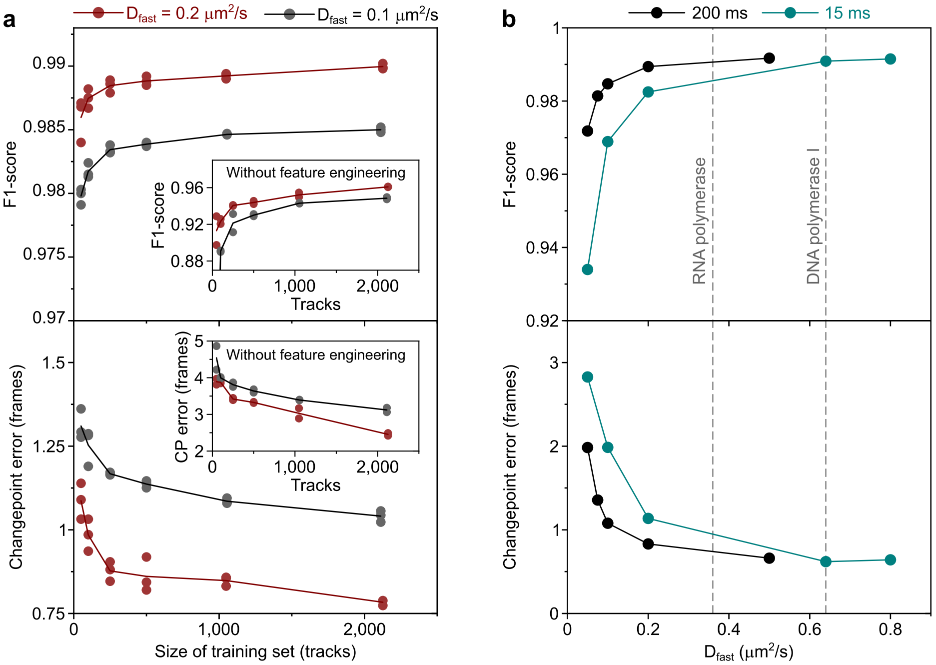

Fig. 7 | Classification performance of DeepTRACE models with varying training set size and separation of diffusion coefficients. a F1-score (top) and mean changepoint error (bottom) as a function of training set size for simulations of timelapse imaging using intervals of 200 ms, with a slow state diffusion coefficient of 0.01 μm²/s, and a fast state diffusion coefficient of 0.1 μm²/s (burgundy circles) or 0.2 μm²/s (grey circles). The burgundy and black lines represent means of *n* = 3 independently trained models. Shown inset are the same metrics obtained using identical parameters, model architecture, hardware, training data, and evaluation data, but eliminating the feature engineering process by restriction to only raw tracking data (x, y, time), performed with *n* = 2 independently trained models; all individual model performances are shown. For each model, the training and validation split, shuffling, random track subsampling, and batch construction processes were performed independently. For partial datasets (n ≤ 500 independent tracks), the original data were downsampled through random track elimination independently prior to splitting the training and validation data. **b** Variation of F1-score (top) and mean changepoint error (bottom) as a function of diffusion coefficient of the simulated fast state, with the slow state fixed at 0.01 μm²/s. The panels show performance from simulated timelapse imaging with an interframe time of 200 ms (black circles), and 15 ms (teal circles) common with experimental techniques such as trackingPALM microscopy. For inter-frame times of 200 ms models were trained using n = 1065, 1,048, 1054, 1048, and 1066 independent tracks for simulated diffusion coefficients with a fast state of D_{fast} = 0.05, 0.075, 0.1, 0.2, and 0.5 μm²/s respectively. For inter-frame times of 15 ms models were trained using *n* = 1110, 1167, 1081, 1098, and 1104 independent tracks with a fast state of D_{fast} = 0.05, 0.1, 0.2, 0.64, and 0.8 μm²/s respectively. Vertical dashed reference lines indicate established diffusion coefficients of the target searching ('mobile') state for two DNA-binding proteins known to exhibit substantial slowdown due to intra-frame transient interactions with the chromosome (bacterial RNA polymerase[11] and DNA polymerase I[16]). All models were evaluated on data obtained from separate simulations, completely independent from the training data.

Two-state simulations consisted of a slow state with a fixed diffusion coefficient of 0.01 μm²/s, and a fast state that was varied between simulations across the range 0.05 μm²/s to 0.8 μm²/s. Each molecule began at a random position within the cell cytoplasm, undergoing Brownian motion, and confined by the cell membrane with reflective boundaries. State occupancies began in equilibrium, with two-thirds of molecules beginning in the slow state. For time-lapse imaging, transition rates between diffusive states were set at 0.1 Hz and 0.05 Hz for fast-to-slow and slow-to-fast transitions respectively; an asymmetry that simulated class imbalance. For simulations of continuous imaging, the transition rates were adjusted to 0.667 Hz and 1.333 Hz to produce a similar density of observed changepoints in the faster photobleaching regime. This scaling of transition rates reflects the increasingly common use of temporal patterning techniques and improved probe stability that enables the experimentalist to adjust the imaging timescale to capture reasonable densities of transitions (e.g. extending dark times for slower processes). These class transition rates were varied only for simulations investigating the impact of experimental perturbation on model performance (Supplementary Fig. 9a); in which class transition rates under continuous imaging were reduced to 0.1 Hz and 0.05 Hz under continuous imaging for the evaluation dataset.

(ii) Simulations of reversible binding to a slow diffusing polar-localised plasmid

A molecule was simulated with a diffusion coefficient of 0.2 μm²/s diffusing freely within the cell cytoplasm, and able to bind reversibly to a polar-localised plasmid within each end cap (a region defined as 200 nm from the cell pole). Within these regions the molecule experienced a binding probability equivalent to an on-rate of 0.1 Hz. Once in the bound state, the molecule was confined to the polar region, with diffusion coefficient 0.01 μm²/s, until released from the plasmid with an off rate of 0.05 Hz, and transitioned back to the initial diffusive state in which it was again able explore the entire cytoplasm. This process was allowed to repeat multiple times before probe photobleaching. Simulations were initiated in equilibrium, with 4.07% of molecules initially in the polar-bound state with the remainder beginning in the cytoplasm; a fraction that accounted for both transition rates and the relative volume fraction within which transitions to the polar state can occur. In each case molecule start locations were drawn from a uniform distribution within the permitted volume regions for each class.

(iii) Simulations of irreversible transport across the periplasmic membrane

Transport across the membrane separating the cytoplasm and periplasm (a 20 nm-thick shell at the cell periphery enclosing the cytoplasm with reflective boundaries) was simulated for a single molecule initially undergoing free diffusion within the cytoplasm with a diffusion coefficient of 0.2 μm²/s. Collision with the membrane resulted in a probability of transmembrane transport set by an ‘absorption rate’ of 0.001 Hz, causing it to become immobilised at the closest point of contact. The molecule was otherwise reflected at the boundary. Following transport, the molecule was transferred irreversibly to the periplasmic space, which it continued to explore with a diffusion coefficient of 1.0 μm^2/s.

Single molecule localisation and tracking

Candidate single molecule locations were obtained from spatial bandpass filtering of each fluorescence video frame, using a threshold tuned to the imaging conditions. Super-resolution localisation was then performed through 2D elliptical Gaussian fitting of candidate positions, producing several features (e.g. peak intensity, width of Gaussian fit major axis) that were incorporated into DeepTRACE’s raw feature set.

Localisations were connected into tracks using a nearest-neighbour search between neighbouring frames within a maximum distance of 7 pixels (672 nm). Tracking was performed independently within each segmented cell boundary. For simulated data the maximum connecting distance for tracking was extended to 50 pixels (4.8 μm) as false track connections from temporal overlap were not possible due to the sequential nature of the simulations. A tracking memory parameter of one was used, allowing track connection to bridge single frame blinking events.

Track filtering

Tracks imported into DeepTRACE were filtered with a minimum track length of 50 and 65 frames for simulated and experimental data respectively. After filtering for overall length, tracks were processed using DeepTRACE’s ‘truncation by localisation filtering’ option with a tolerance of two localisations, resulting in any tracks that coincide with more than two localisations (regardless of whether they were tracked) being truncated to eliminate any temporal overlaps for which track history is uncertain. This information-preserving approach can result in tracks shorter than the minimum track length appearing in the dataset.

DeepTRACE graphical interface

All analysis was performed using DeepTRACE’s cross-platform Graphical User Interface (GUI). The GUI supports data pre-processing, feature engineering and selection, human annotation, model construction and training, track segmentation, and a wide range of downstream analytics tools.

Feature engineering

Data preprocessing by DeepTRACE engineered 60 features (Supplementary Data 2). These features have varying uses in different experimental contexts. We group these features broadly into four main categories,

- Raw Features: Features taken directly from the source data (localisation fitting parameters, photometric properties from images, and tracking data), such as peak intensity, x-y coordinates, or local background noise.
- Static Engineered Features: Features that are engineered without temporal context, and depend only upon the current localisation or two points immediately flanking the current localisation. Examples include step size, distance to the nearest cell membrane, and step angle.
- Local Window Features: Features that are computed from a local context window, such as smoothed step size, step angle asymmetry, or path straightness.
- Delta Features: Features that are computed from comparison of two local context windows, immediately preceding and following the datapoint. These features are particularly powerful for the identification of changepoints, especially when implemented for small models and those without native temporal context. In larger models, these simplified representations of temporal behaviour accelerate learning by shortcutting the process of constructing higher order representations of changepoints from raw features. Delta features are computed as either the difference or the ratio between the window pair, with the preceding window defined as the denominator or subtrahend. Examples include Δdispersion, mean step size ratio, and centroid displacement.

As window size is a trade-off between accuracy and sensitivity to changepoints, for all local features except for smoothed step size we used a local window size of 11 frames (five localisations flanking the classified localisation); this scale was obtained from empirical testing with a binary search. All delta features used a pair of windows each containing four frames, with the evaluated localisation being the first datapoint of the second temporal window.

Close to the track end points in which one frame becomes compressed, the value for each feature is scaled appropriately where possible by the available number of accessible frames, trading increased noise in exchange for improved temporal sensitivity and access to end point proximal regions. For some static features it is not possible to scale values when windows are compressed close to end points (e.g. relative step angle, which requires a minimum of three localisations). As data imputation in many experimental tasks can produce errors arising from temporal context, DeepTRACE instead sets these values to zero, and we used options accessible within the GUI to truncate the first and final two localisations prior to training, as described in the section ‘Preparation of training data’.

We provide here formal definitions for non-intuitive features, and conceptually simple features which may have ambiguous definitions such as smoothed step size. Remaining features for which the definition is obvious (such as step size which computes the simple Euclidean distance between frames, or time step which computes the temporal separation between localisations to learn behaviours associated with single-molecule blinking events) are listed separately.

Centroid displacement

Centroid displacement is a measure of the Euclidean distance between centroids of localisations in windows either side of the localisation being evaluated. Specifically, centroid displacement (d_C) was defined as,

$$d_C = ||\bar{\mathbf{X}}_B - \bar{\mathbf{X}}_A|| \tag{1}$$

Where $\bar{\mathbf{X}}$ represents the centroid of all W localisations within the preceding (A) and following (B) windows, defined as follows,

$$\bar{\mathbf{X}} = (\bar{x}, \bar{y}) = \left(\frac{1}{W}\sum_{j=1}^{W} x_j, \frac{1}{W}\sum_{j=1}^{W} y_j \right) \tag{2}$$

Delta dispersion

Delta dispersion quantifies changes in the spatial spread of localisations between two adjacent temporal windows. We defined the dispersion ψ of a window containing W positions $X_i = (x_j, y_j)$ as the mean Euclidean norm per point, summed over all points, relative to the window centroid,

$$\psi = \frac{1}{W}\sum_{i=1}^{W} ||\mathbf{X}_i - \bar{\mathbf{X}}|| \tag{3}$$

Delta dispersion and dispersion ratio were then computed by the difference or ratio of ψ within each window.

Smoothed step size

Smoothed step size is particularly useful for suppressing localisation error, statistical noise, and projection effects. DeepTRACE provides multiple smoothing methods, including Gaussian kernel smoothing, moving mean, local regression such as (R)LOWESS and (R)LOESS, and the Savitzky-Golay filter. For all work presented here, a median filter with a window size of 3 was used.

Longitude and latitude

To map localisations onto a universal model-cell coordinate system, we perform a transformation along the major and minor axes, which we term longitude (L), and latitude (ℓ). The longitude feature is obtained from the contour length of the projected position along the cell midline, while latitude is computed as the relative distance between the projected midline location and cell boundary. The coordinate system is designed such that the origin is placed at the mid-point of the midline contour, with the boundary at ±0.5.

The midline vertices, $\boldsymbol{M}_i$, are obtained by averaging each left and right coordinate pair from the MicrobeTracker-formatted boundary,

$$\boldsymbol{M}_i = \frac{1}{2}\left(\boldsymbol{E}_i^L + \boldsymbol{E}_i^R\right) \tag{4}$$

Where $\boldsymbol{E}_i^L$ and $\boldsymbol{E}_i^R$ represent the paired vertices defining each edge of the segmented cell boundary. Each localisation is projected onto the closest midline segment $\left[\boldsymbol{M}_i, \boldsymbol{M}_{i+1}\right]$, and longitude is computed as,

$$L = \lambda||\boldsymbol{M}_{i+1} - \boldsymbol{M}_i|| + \sum_{j=1}^{i-1} ||\boldsymbol{M}_{j+1} - \boldsymbol{M}_j|| \tag{5}$$

Where λ is the fractional position along the segment,

$$\lambda = \frac{\left(\boldsymbol{X} - \boldsymbol{M}_i\right) \cdot \left(\boldsymbol{M}_{i+1} - \boldsymbol{M}_i\right)}{||\boldsymbol{M}_{i+1} - \boldsymbol{M}_i||^2} \tag{6}$$

Finally, we define latitude as the relative perpendicular distance from the midline projection **T** to the localisation **X**, normalised by the distance to the intersection point **F** of the line **TX** with the cell boundary,

$$\ell = \frac{||\boldsymbol{X} - \boldsymbol{T}||}{||\boldsymbol{F} - \boldsymbol{T}||} \tag{7}$$

Note that both latitude and longitude can exceed the cell boundary if the input tracking data associates such localisations with the segmented cell; this enables DeepTRACE to retain tracking of localisations that fall outside of the segmented boundary due to localisation uncertainty or drift, and to process nearby contextual data, while also being able to exclude such data when not desired, for example in constructing spatial maps.

Local straightness

Path straightness is a geometric local feature quantifying the directness of the molecule's motion. Straightness was defined as the ratio of the distance between start and end points of the windowed track to its full contour length,

$$S = \frac{||\boldsymbol{X}_W - \boldsymbol{X}_1||}{\sum_{i=1}^{W-1} ||\boldsymbol{X}_{i+1} - \boldsymbol{X}_i||} \tag{8}$$

Local path efficiency

Efficiency is a normalised version of path straightness. We used the definition of efficiency put forward by Kowalek et al.[28]; for a local window containing W localisations, path efficiency was computed as the ratio of the squared total displacement (between window start and end points) to the product of the number of steps and the sum of the squared step lengths,

$$E = \frac{||\boldsymbol{X}_W - \boldsymbol{X}_1||^2}{(W-1)\sum_{i=1}^{W-1} ||\boldsymbol{X}_{i+1} - \boldsymbol{X}_i||^2} \tag{9}$$

$E \to 1$ for directed motion, and $E \to 0$ for extremely convoluted paths. The normalisation of path efficiency makes it more resistant to compressed data sampling regions, such as towards the ends of tracks. While dynamically compressing the local window increases noise, it avoids reporting on an artificial increase in the directness of the path towards the ends of the track and in regions of excessive photoblinking.

Local fractal dimension

We used the definition of fractal dimension (D_f) provided by Katz and George[29] as follows,

$$D_f = \frac{\log(W-1)}{\log(W-1) + \log\left(\frac{d}{L}\right)} \tag{10}$$

Where W is the total number of localisations in the window, d is the maximum distance between any two points in the window, and L is the total contour length. The feature has been shown to be effective at discriminating mobility modes.

Local trappedness

Following Saxton et al. [30] trappedness was defined as,

$$p_t = 1 - \exp\left(c_1 - c_2 \frac{Dt}{r_0^2}\right) \tag{11}$$

Where D is the diffusion coefficient obtained from a linear fit to the mean squared displacement with a maximum lag time, t, of two frames, r_0 defines the trapping region represented by half of the maximum step size along the path, and c_1 and c_2 are empirical fitting constants adjusted accordingly for our experiments.

Local step angle asymmetry

Step angle asymmetry (A_s) is a proxy for the degree of spatial confinement. Expanding on the definition from Izeddin et al.[31], we defined the step size asymmetry as,

$$A_s = \log_2\left(\frac{N_{\text{fwd}} + 1}{N_{\text{bwd}} + 1}\right) \tag{12}$$

Where N_{fwd} represents the number of forward steps with step angles between 0 and 30 degrees, and N_{bwd} represents the number of backward steps with step angles between 150 and 180 degrees. Values for A_s are negative for backward step angle biases, and positive for forward biases. We retained the $\log_2$ notation for consistency with past work[31] despite the rescaling of this feature to a symmetric distribution around zero being unnecessary due to the Z-score feature scaling internal to DeepTRACE. The introduction of $+1$ here improves robustness of the metric by preventing division or multiplication by zero when operating on small local windows.

Local kurtosis

The gyration tensor for a set of points $\boldsymbol{X}$, within a window containing W localisations was defined as,

$$\boldsymbol{G} = \frac{1}{W}\sum_{i=1}^{W}\left(\boldsymbol{X}_i - \bar{\boldsymbol{X}}\right)\left(\boldsymbol{X}_i - \bar{\boldsymbol{X}}\right)^T = \begin{bmatrix} G_{xx} & G_{xy} \\ G_{yx} & G_{yy} \end{bmatrix} \tag{13}$$

The dominant eigenvector, representing the largest spread of localisations, was obtained by diagonalising $\boldsymbol{G}\boldsymbol{v} = \lambda\boldsymbol{v}$, where λ are the eigenvalues, and $\boldsymbol{v}$ the eigenvectors defining the principal axis. The projection, p_i, of each point along the dominant eigenvector $\boldsymbol{v}_{\max}$ was obtained by,

$$p_i = \left(\boldsymbol{X}_i - \bar{\boldsymbol{X}}\right) \cdot \boldsymbol{v}_{\max} \tag{14}$$

The local kurtosis is then computed from these projected positions as,

$$K = \frac{1}{W}\sum_{i=1}^{W}\left(\frac{p_i - \bar{p}}{\sigma_p}\right)^4 \tag{15}$$

Where σ_p represents the standard deviation of positions projected along the dominant eigenvector.

Local step size kurtosis

Step size kurtosis is a measure of tailed-ness of the step size distribution, defined as,

$$K_s = \frac{1}{W-1}\sum_{i=1}^{W-1}\left(\frac{s_i - \bar{s}}{\sigma_s}\right)^4 \tag{16}$$

Where $s_i = ||\boldsymbol{X}_{i+1} - \boldsymbol{X}_i||$ is the size of the i^{th} step, $\bar{s}$ is the mean step size, and σ_s is the standard deviation of step sizes between the W localisations within the local window,

$$\sigma_s = \sqrt{\frac{1}{W-1}\sum_{i=1}^{W-1}\left(s_i - \bar{s}\right)^2} \tag{17}$$

Local maximal excursion

Local maximal excursion (ME) quantifies the extent of large, isolated displacements within a local window relative to the net displacement over that window. It is particularly effective at detecting transient high-mobility events, such as sudden or transient escape from confinement. The local maximal excursion for a window centred at localisation i, was computed as the ratio of the maximum Euclidean distance between consecutive points within the window to the net displacement across the window,

$$\mathrm{ME} = \frac{\max ||\boldsymbol{X}_{i+1} - \boldsymbol{X}_i||}{||\boldsymbol{X}_W - \boldsymbol{X}_1||} \tag{18}$$

Where $\boldsymbol{X}_1$ and $\boldsymbol{X}_W$ are the spatial coordinates of the molecule in the first and last frame of the local window respectively. When combined with features that encode local diffusion coefficient estimates, the model can be particularly effective in identifying rapid or brief transitions in diffusive state.

Local velocity autocorrelation function (VACF)

The local velocity autocorrelation function (V) quantifies the temporal correlation of motion over short time scales, capturing the tendency of the molecule to maintain its direction of movement. It was computed as the average dot product between a step vector and a step vector offset by fixed temporal distance (n frames) relative to itself, as follows,

$$V = \frac{1}{W-n-1}\sum_{i=1}^{W-n-1}\left(\boldsymbol{X}_{i+1+n} - \boldsymbol{X}_{i+n}\right) \cdot \left(\boldsymbol{X}_{i+1} - \boldsymbol{X}_i\right) \tag{19}$$

Throughout this work we used $n = 1$. Positive values indicate persistent motion (the molecule tends to continue in the same direction), while negative values indicate anti-persistent motion (the molecule tends to reverse direction). Values near zero correspond to random motion. The local velocity autocorrelation function is particularly useful for assisting models in identifying changes in diffusion type.

Spot size and area

Spot size (ϕ) and spot area (A) quantify the spatial extent of localisations which encodes information about intra-frame particle motion, particularly in continuous exposure experiments. These features were computed from the standard deviation of the major and minor axes (σ) of the 2D elliptical Gaussian fits during the localisation process,

$$\phi = 2\sqrt{\sigma_{\mathrm{major}}^2 + \sigma_{\mathrm{minor}}^2} \tag{20}$$

$$A = \pi\sigma_{\mathrm{major}}\sigma_{\mathrm{minor}} \tag{21}$$

Step angle features

Three step angle features were engineered: the relative step angle between successive steps (θ_{rel}), the step angle relative to the image (θ_{img}), and the step angle relative to the major cell axis (θ_{cell}), using the step vectors at positions $i-2$ to i,

$$\theta_{\mathrm{img}} = \tan^{-1}\left(\frac{\left(\boldsymbol{X}_{\mathrm{i}} - \boldsymbol{X}_{\mathrm{i}-1}\right)_y}{\left(\boldsymbol{X}_{\mathrm{i}} - \boldsymbol{X}_{\mathrm{i}-1}\right)_x}\right) \tag{22}$$

$$\theta_{\mathrm{cell},i} = \tan^{-1}\left(\frac{||\left(\boldsymbol{X}_i - \boldsymbol{X}_{i-1}\right) \times \hat{e}_{\mathrm{cell}}||}{\left(\boldsymbol{X}_i - \boldsymbol{X}_{i-1}\right) \cdot \hat{e}_{\mathrm{cell}}}\right) \tag{23}$$

$$\theta_{\mathrm{rel}} = \cos^{-1}\left(\frac{\left(\boldsymbol{X}_i - \boldsymbol{X}_{i-1}\right)'_y}{\left(\boldsymbol{X}_{\mathrm{i}} - \boldsymbol{X}_{\mathrm{i}-1}\right)'_x}\right) \tag{24}$$

Where $\hat{e}_{\mathrm{cell}}$ is a unit vector along the major cell axis, and $\left(\boldsymbol{X}_{\mathrm{i}} - \boldsymbol{X}_{\mathrm{i}-1}\right)'$ represents the step vector from the current step rotated onto the frame of the previous step using the transformation $\boldsymbol{X}'_i = \boldsymbol{R}_{i-1}\left(\boldsymbol{X}_i - \boldsymbol{X}_{i-1}\right)$; the rotation matrix $\boldsymbol{R}_{i-1}$ is defined as,

$$\boldsymbol{R}_{i-1} = \begin{bmatrix} \cos\theta_{\mathrm{img},\,i-1} & -\sin\theta_{\mathrm{img},\,i-1} \\ \sin\theta_{\mathrm{img},\,i-1} & \cos\theta_{\mathrm{img},\,i-1} \end{bmatrix} \tag{25}$$

Note that for relative step angle, the previous angle ($\theta_{\mathrm{img},i-1}$) requires the localisation in frame ($i-2$) to compute the previous step vector, and as a result the relative step angle feature contains two zero values at the start of each track due to insufficient data.

Apparent diffusion coefficient

Using a mean squared displacement (MSD) matrix compiled for all pairwise displacements within the window, the local apparent diffusion coefficient (D^*) feature was computed from a linear fit to,

$$D^* = \frac{\mathrm{MSD}}{4\tau} \tag{26}$$

Where τ is the lag time. For all models trained here, we chose to use a maximum lag time equal to the window size to simplify the definition of local features, something not recommended in most contexts. This approach differs significantly from our formal calculation of D^* used for global diffusion analysis in which calculation is performed using $\tau_{\max} = 2$.

Feature selection

Feature selection was performed in five stages,

1. Initial feature selection based on experimental domain knowledge.
2. Visual screening of class-conditional feature distributions and class separability metrics using DeepTRACE's Feature Visualisation tool to identify the static discriminatory power of each feature.

3. Feature ranking via feature-class mutual information using DeepTRACE's Feature Ranking tool.
4. Redundancy reduction through pairwise Pearson correlation thresholding ($|r| < 0.9$), and identification of non-linear dependencies using mutual information heatmaps, both computed using DeepTRACE's Feature Ranking tool.
5. Final selection used permutation importance measured on a candidate model in changepoint proximal and distal regions, via DeepTRACE's Permutation Importance tool.

Redundant features were removed iteratively based on feature-class mutual information, permutation importance, and class separability metrics, in order of increasing utility until no feature redundancies remained.

In practice, because most of our experiments involved systematic comparisons of model performance under varying conditions (e.g., training set size, separation of diffusive states, experimental perturbation), it was critical to minimise variability introduced by the feature selection itself arising from a large input feature space; we therefore further constrained the feature set to a small, high-confidence subset to ensure stable and reproducible learning dynamics across runs, resulting in fairer comparisons.

DeepTRACE's SHAP-based interpretability tools and interactive 3D feature scatter plots were not used directly for feature selection.

Class-conditional feature distribution plots

To assess the class separability of each feature in the absence of temporal context, we used DeepTRACE's Feature Visualisation tool, which provided a visual indication of class separability using histograms and kernel density estimates (Fig. 2a), together with various quantitative metrics described below. Histograms were constructed by adaptive binning of feature distributions into 25 bins with edges computed globally across all tracks and classes to ensure reliable sampling despite extreme differences in feature distributions, scenarios with large class imbalance, and small datasets.

The two-sample Kolmogorov-Smirnov statistic (D_{KS}) between the empirical cumulative distribution functions of feature values for each class, which can be computed directly on raw feature data without binning, was defined as,

$$D_{\mathrm{KS}} = \sup_x \left|F_{1,n}(x) - F_{2,m}(x)\right| \tag{27}$$

Where F_1 and F_2 represent the cumulative distribution functions of the classes being compared, containing n and m samples respectively. The 1st order Wasserstein distance (W_1) was computed from adaptively binned data as,

$$W_1 = \sum_i^N \left|F_1(i) - F_2(i)\right| \cdot \Delta x_i \tag{28}$$

Where N is the number of bins, and Δx_i is the width of the i^{th} bin. Finally, the more visually intuitive histogram-based metric (Ω) computed the fractional non-overlap between classes, defined as,

$$\Omega = 1 - \frac{\sum_i^N \min\left(p_1(i), p_2(i)\right)}{\sum_i^N \max\left(p_1(i), p_2(i)\right)} \tag{29}$$

Where $p_1(i)$ and $p_2(i)$ are occupancies of the i^{th} bin in class 1 and 2 respectively.

Pairwise Pearson Correlation

To identify redundant features for removal, linear correlations were computed using DeepTRACE's Feature Ranking tool. The pairwise Pearson correlation between every feature pair in the dataset was computed across all tracks using,

$$r_{xy} = \frac{\sum_{i=1}^N \left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)}{\sqrt{\sum_{i=1}^N \left(x_i - \bar{x}\right)^2} \cdot \sqrt{\sum_{i=1}^N \left(y_i - \bar{y}\right)^2}} \tag{30}$$

Where x_i and y_i represent the i^{th} values of two features, $\bar{x}$ and $\bar{y}$ are their mean values, and N is the total number of localisations in the combined dataset.

The resulting correlation matrix was visualised as a heatmap (Supplementary Fig. 1a), enabling identification of strong linear correlations which indicates high redundancy.

Mutual information

Feature importance was assessed via the mutual information shared by each feature with the class assigned from human annotation or ground truth. To account for the wide diversity of feature distributions, including those with extreme variations in kurtosis (heavy tails or sharp peaks), we discretise the data using adaptive binning into 25 equally-populated bins, such that each distribution is sampled at intervals of even density throughout the dataset. We then compute the mutual information metric as follows,

$$\mathrm{MI} = \sum_{i=1}^{25} \sum_{j=1}^{M} P(X_i, Y_j) \cdot \log_2\left(\frac{P(X_i, Y_j)}{P(X_i) \cdot P(Y_j)}\right) \tag{31}$$

Where X represents the discretised feature, Y is an integer assigned to each of M classes, $P(X_i, Y_j)$ is the joint probability mass function of X and Y, and $P(X_i)$ and $P(Y_j)$ are the marginal probability mass functions of the feature X and class Y.

Pairwise mutual information heatmaps were generated by computing the mutual information between all feature pairs (X_i, Y_j), where Y in this case represents the second feature.

Preparation of training data

All features were standardised using Z-score normalisation globally across all tracks prior to splitting training, validation, and test data. The Z-score transform rescaled the data to a mean of zero, and standard deviation of one, such that for a given feature x, each datapoint x_i was transformed to its Z-score z_i as,

$$z_i = \frac{x_i - \mu_x}{\sigma_x} \tag{32}$$

Where μ_x and σ_x represent the global mean and sample standard deviation of the feature x respectively,

$$\mu_x = \frac{1}{\mathrm{n}} \sum_{i=1}^{n} x_i \tag{33}$$

$$\sigma_x = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} \left(x_i - \mu_x\right)^2} \tag{34}$$

and n is the number of timepoints across all tracks in the dataset. While DeepTRACE offers alternative forms of data normalisation, throughout this work we use Z-score as this provides the most robust handling for feature sets with widely differing distributions. These parameters used were stored with the trained model, ensuring consistent preprocessing during evaluation of unseen data.

To enforce sample independence, data shuffling was performed at the individual track level prior to splitting into training, validation, and optional test data. A split of 85% training and 14% validation data was used, and a small test split (1%) was discarded due to the current internal configuration of the training framework; however, all test results reported in this study were obtained from completely separate experimental and simulated datasets, providing a truly independent and more robust assessment of model generalisation than a simple split of the training data. Drawing test data from separate experiments was also necessary as single-molecule tracking data often contain information and artefacts specific to the recording (e.g., environmental conditions, sample heterogeneity, hardware variability, or optical effects such as focal plane positioning).

Given the small dataset sizes, sequences used for training were truncated by two timepoints at both ends to ensure only regions with complete feature coverage were used for training, accelerating the learning process by avoiding learning special cases related to end point proximity.

DeepTRACE offers a range of subsampling methods to compile training data; throughout this work we use the random subsampling method with a temporal context window of 25 localisations. DeepTRACE performed subsampling by computing the maximum number of overlapping subtracks in the shortest track of the dataset, and used this number to randomly subsample without replacement all tracks into the same number of examples. This unusual temporal augmentation strategy increases sample diversity by exposing each datapoint to the largest possible range of local contexts while ensuring that equal weighting was provided to each track regardless of length, maximising the impact of limited data. Resulting examples were then shuffled randomly into batches after separation at the whole-track level of the training and validation data. Note that the concept of training epoch varies slightly with this augmentation approach as unique short sequences appear multiple times in time-shifted contexts across different batches.

Model selection

While DeepTRACE supports a range of models, including recurrent neural networks (RNNs) and ensemble-based methods, throughout this study we chose Bidirectional Long Short-Term Memory networks (BiLSTMs) followed by a self-attention mechanism. Throughout early testing this configuration consistently outperformed Gated Recurrent Unit (GRU) and ensemble decision tree models with feature-engineered temporal context.

Additional model types remain available in DeepTRACE for applications beyond the scope of this study, including random forests for analysis of processes lacking temporal dependencies (e.g. fixed cell experiments), and model interpretability tasks.

We conducted empirical benchmarking of different configurations, including models with deeper RNN stacks and multi-head attention mechanisms, but observed no consistent improvement on the small datasets targeted by this study. While the DeepTRACE GUI supports flexible construction of all such models based around a modular architecture (Supplementary Fig. 10), all models presented here use a single BiLSTM layer and single attention head. This design prioritises reduced complexity, rapid training on small-scale datasets, and compatibility with limited computing resources, to align with DeepTRACE's goal of broad accessibility across the single-molecule tracking community.

Unless otherwise stated, we used BiLSTM layers with 100 hidden units (1.6×10^5 parameters), a post-RNN dropout layer with 50% dropout, input and recurrent L2 regularisation factors of 2×10^{-5} and no L2 penalty on biases. We also found that substantially smaller models (10–15 hidden units; $< 5 \times 10^3$ parameters) can achieve similar performance across all tasks, but often require multiple training runs to do so. This behaviour is consistent with the lottery ticket hypothesis[32,33], which suggests that larger networks are more likely to contain well-initialised sparse subnetworks (referred to as 'winning tickets') that train reliably from a single run. For this reason, DeepTRACE intentionally adopts mildly over-parameterised defaults to provide much more reliable training for users while having negligible impact on training time, inference speed, or hardware requirements.

Initial hyperparameter ranges were explored with a custom-built, Bayesian-inspired search implemented during early development, and final configurations were selected via manual tuning. BiLSTM layers used gating mechanisms (tanh for state updates and sigmoid for gating) that are suited to learning long-range patterns in time series data.

The training process

Training used the Adam optimiser with piecewise learning rate scheduling. The initial learning rate was set to 10^{-4} and reduced to 30% of the previous rate every five epochs. All models were trained with mini-batches of 16 examples. Validation loss and accuracy were monitored using a holdout validation set evaluated every 500 iterations. In order to prevent overfitting to the training data, early stopping was triggered following 10 consecutive evaluations without improvement, or after a maximum of 50 epochs. When working with datasets containing fewer than 250 tracks, and for all models in the experimental perturbation simulations, the validation patience and maximum epochs were extended to 15 consecutive evaluations, and 100 epochs respectively. The final model weights are those that produced the lowest validation loss. Final model evaluation was performed on a completely independent test set from separately-run simulations and experiments.

Hardware

Models for the reversible two-state diffusion simulations (Figs. 3b,c, 6) were trained using a single core of a typical laptop single CPU (M2 Macbook Air, 2022) in macOS. All other models were trained using a low-cost GPU (Nvidia Quadro P1000, released February 2017, 4 GB GDDR5 memory, 82 GB/s bandwidth, with no Tensor Cores and no accelerated FP16 support) on a typical Windows desktop PC while running other background tasks. This low-powered, dated GPU was chosen intentionally to demonstrate method efficacy on modest hardware; by today's standards, this GPU (designed for lightweight CAD tasks) is outdated and lacks the computational power and memory required for large-scale deep learning workloads.

Loss functions

The total loss was computed as the contribution $L_{b,t}$ for all timepoints, T, across the mini-batch B,

$$\text{Total Loss} = \sum_{b=1}^{B} \sum_{t=1}^{T} L_{b,t} \tag{35}$$

The class-weighted loss was computed using cross-entropy with weights obtained from the inverse of normalised class frequency, to obtain the loss per localisation instance as follows,

$$L_{\text{class},b,t} = -\sum_{c=1}^{C} w_c \, y_{c,b,t} \, \log\left(p_{c,b,t}\right) \tag{36}$$

Where w_c is the class weight for class c; $y_{c,b,t}$ is the true label (1 or 0) for class c, one-hot encoded at batch index b, and time point t; and $p_{c,b,t}$ is the predicted probability.

We also defined a changepoint-masked loss function that computes class-weighted cross-entropy loss for a single data point (indexed by b, t, and c) with an extra weight multiplier for changepoint-proximal regions. Changepoints were defined as any change in ground truth label in adjacent localisations (i.e. $y_{b,t} \neq y_{b,t-1}$). The combined class-weighted, and changepoint-mask weighted loss contribution for per localisation instance was defined as,

$$L_{b,t} = -\sum_{c=1}^{C} \left(1 + F\, m_{b,t}\right) w_c \, y_{c,b,t} \log\left(p_{c,b,t}\right) \tag{37}$$

Where F is the additional weighting assigned to localisations indexed by the changepoint mask; $m_{b,t}$ is the entry from the binary changepoint mask at batch index b and time point t. Throughout this work we primarily employ the class-weighted loss as most applications concern the extraction of global biophysical parameters from analysed tracks, which was found to be relatively insensitive to minor increases in changepoint error.

Inference

DeepTRACE supports several inference modes, including whole-track and segment-based classification; however, all results presented in this study use its sliding window method. Each track was divided into overlapping temporal windows, and the trained model was applied independently to each window to predict class probabilities using sequence-to-sequence classification. Since localisations appear in multiple overlapping windows, DeepTRACE computes a consensus classification by averaging the softmax probabilities across all windows in which each localisation appears. This allows each annotation to reflect multiple temporally shifted contexts, improving robustness to local noise. The final assigned class corresponds to the class with the highest mean probability.

In addition to the predicted class, DeepTRACE also retains the full probability distribution and number of contributing windows for each localisation, enabling confidence-aware post-processing (e.g. Gaussian or exponential smoothing, enforcing minimum time per state, or enabling state reassignment based on permitted event sequences) if desired. However, to faithfully evaluate the raw performance of each model, no post-processing was applied as this may artificially boost the reported performance, and we instead report on raw performance metrics which reflect the underlying model classification.

Visual verification of model performance

To avoid operating as a black box, DeepTRACE's classification performance was monitored via multiple evaluation metrics, discrete statistics, comparison to human and ground truth data, and visual interrogation of individual tracks. Following segmentation, DeepTRACE was used to perform extensive downstream analysis of segmented tracks, including mobility analysis, spatial mapping, descriptive statistics, and event rates.

Performance metrics

Classification performance was quantified using mean changepoint error and various macro-averaged metrics. If *TP*, *FP*, *TN*, and *FN* are the number of true positives, false positives, true negatives, and false negatives respectively, then we compute each metric using standard definitions as follows,

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{38}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{39}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{40}$$

$$\text{Jaccard} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \tag{41}$$

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{42}$$

Macro-averaged F1-score was computed as the mean F1-score across all classes.

The mean changepoint error was defined as the mean difference in number of frames between matched changepoints of the same type in the model classifications and ground truth within a maximum distance of five frames, as follows,

$$E_{\text{CP}} = \frac{1}{N}\sum_{i=1}^{N}\left|t_{\text{pred}} - t_{\text{gt}}\right| \tag{43}$$

Unmatched changepoints are assigned the maximum changepoint error. Changepoint metrics for real experimental data represent the difference between model annotations and changepoints in the human annotations.

Permutation importance

Permutation importance was evaluated using DeepTRACE's Permutation Importance tool, which computed the drop in macro-averaged accuracy of the model across all classes when each feature is permuted. For every feature used by the model, localisation values across all tracks were shuffled, breaking their association with the class labels. Model accuracy was then re-evaluated, and the mean drop in performance across n = 5 permutations was recorded.

DeepTRACE computed the permutation importance separately within the changepoint-proximal and changepoint-distal regions (defined as the four localisations closest to each ground truth changepoint), and ranked features by the sum of their importance across both regions to identify features most heavily influencing the model. This changepoint masking was particularly instructive as some engineered features were specifically designed for changepoint identification and may not exhibit strong class separation in distal regions, which constitute the majority of frames in most single-molecule tracking data. As with other metrics, ground truth for real experimental data was taken as the human annotations.

SHapley Additive exPlanations (SHAP) values

SHAP values (Supplementary Fig. 1c) were computed using Random Forest surrogate model with MATLAB's Tree SHAP algorithm.

Spatial mapping

We used DeepTRACE's Spatial Mapping tool to generate 2D heatmaps (Fig. 4c inset), STORM-like reconstructions (Supplementary Fig. 2b,c), and cell-axial projections of classified molecule localisations. Localisations were transformed into model cell coordinates using the longitude and latitude features engineered by DeepTRACE, with those falling outside the segmented cell boundary excluded from visualisation. Heatmaps were constructed for each class by binning spatial localisations with a bin size of 25 nm. Reconstructed spatial maps were produced by positioning radially symmetric Gaussian PSFs at each localisation within a model cell, and then rendered by accumulating the contribution from all PSFs into a single, normalised super-resolution image with a pixel size equal to half of the PSF's full width at half maximum (FWHM).

Computation of diffusion coefficients

Diffusion coefficients were computed using DeepTRACE's Diffusion Analysis tool, which compiled all segmented subtracks for each class across the entire dataset. Valid displacement steps were aggregated to construct MSD-lag time relationships for each class, and apparent diffusion coefficients were then calculated from a linear least-squares fit to the first two lag times for each class using $D^* = \text{MSD}/4\tau$. This approach enables robust diffusion analysis without requiring uninterrupted diffusive states, and greatly reduces the excessive truncation-induced noise that limits conventional single-molecule diffusion analysis. While DeepTRACE is able to compile MSD-lag time relationships by truncating the end points of subtracks to further boost performance (by minimising the influence of class averaging present in localisations adjacent to each end point arising from changepoint errors and intra-frame transitions, see 'Simulations' in Methods), we instead compute coefficients using all available steps to more accurately represent the raw segmented tracks, including these sources of error.

Computation of ground truth metrics

Ground truth apparent diffusion coefficients were obtained using standard MSD analysis on complete subtracks segmented using the objective ground truth labels, and computed numerically as a linear fit to lag times of one and two frames as described in 'Computation of diffusion coefficients'. These values therefore contain the same localisation and statistical noise as is present in the tracking data supplied to Spot-On, γMM analysis, and DeepTRACE. Differences between ground truth apparent diffusion coefficients and the simulation input values are a result of a number of sources of noise, including spatial confinement, statistical noise in drawing from distributions (e.g. state transition probability, initial proximity to spatial barrier, step size), projection effects, and localisation noise; the real-world analogues of each exist in experimental data.

Evaluation with Spot-On

To obtain a fair comparison to the widely-used software Spot-On, we analysed the exact set of filtered tracks used by DeepTRACE, reformatted to the Spot-On default format (Frame, Time, Track ID, x, y). Spot-On's two-state model was used for all analysis, with default parameters, which were empirically well-suited to our simulation data. The ranges of D_{bound} and D_{free} were adjusted based on known ground truth values, and bin widths were tuned to improve sampling resolution. Fitting was also performed with both CDF and PDF methods, and using truncated and non-truncated forms to ensure the best possible configuration was obtained.

Gamma mixture model analysis using truncated tracks

Tracks with a minimum length of five localisations were used to compile diffusion histograms, with longer tracks truncated to this minimum length. For each track, the apparent diffusion coefficient D^* was estimated by performing a linear fit to the MSD-lag time relationship and aggregated to populate diffusion histograms.

A two-Gamma mixture model was fit to the ensemble diffusion histogram to estimate the underlying distribution of apparent diffusion coefficients. Following the method of Stracy et al.[11], we fit the probability density function f_D^* of two Gamma distributions with diffusion coefficients D_1 and D_2, and occupancies A_1 and A_2 respectively,

$$f_D^*(x;\, D_1, D_2, A_1, A_2) = A_1 \frac{(4/D_1)^4\, x^3\, e^{-4x/D_1}}{6} + A_2 \frac{(4/D_2)^4\, x^3\, e^{-4x/D_2}}{6} \tag{44}$$

We leave both diffusion coefficients and occupancies as free parameters to avoid biasing the fitting process. To retain consistency with previously published methods, Gamma mixture model analysis was performed on the raw tracking data; as such the analysis contains additional short tracks that were removed by DeepTRACE's track filtering processes.

Segmentation with ResAnDi2

We compared DeepTRACE to ResAnDi2[26], a top-five finalist in both the single-track and ensemble tasks of the 2nd Anomalous Diffusion (AnDi) Challenge. The published pre-trained weights (https://github.com/peardragon/AnDi2) were used without modification. Tracks were cropped or zero-padded to the fixed 200-frame input size of the ResAnDi2 model, and pre-processed with DeepTRACE's automated export tool: translated so the first localisation was at the origin, min–max normalised, and formatted as a single field of view. ResAnDi2's inferred diffusion coefficients (k) were imported into DeepTRACE and mapped to discrete states using fixed thresholding of the predicted k values, rather than using the model's native discrete state assignments. For two-state diffusion simulations, a single midpoint threshold ($k = 0.5$) was applied. For the irreversible transport simulations, two thresholds were used to separate the three mobility regimes. This procedure was applied consistently across datasets.

Construction of diffusion histograms from DeepTRACE-segmented subtracks

Diffusion histograms were constructed with DeepTRACE's Diffusion Histogram tool, which aggregates data from subtracks grouped by class. Each histogram entry corresponds to the apparent diffusion coefficient of a single subtrack computed via linear least squares regression on the first two entries of the mean MSD-lag time relationship, using all available steps within the subtrack.

While all diffusion coefficients presented were computed directly from a single fit to an aggregated MSD-lag time plot for each class (see 'Computation of diffusion coefficients'), coefficients can also be estimated, albeit less accurately, by fitting Gaussian components to each segmented diffusion histogram (Fig. 6d).

Statistics and reproducibility

All evaluation datasets were independent of the datasets used for model training, with no overlap of tracks. For simulated data, separate simulation runs were performed for training and evaluation datasets. For experimental data, training and evaluation datasets were obtained from separate independently prepared samples and were recorded in separate experiments.

Training and validation splits were performed at the track level prior to batch construction and temporal subsampling into training examples, ensuring that no track (in whole or in part) appeared in more than one dataset.

When repeated model training was performed, replicates correspond to independently initialised training runs with independently randomised training/validation splits and batch shuffling. Reported mean values represent arithmetic means across independently trained model instances.

Model performance and comparisons are evaluated using descriptive statistics. No inferential hypothesis testing was performed unless explicitly stated.

Where confidence intervals are shown, these represent nonparametric 95% confidence intervals of the median.

The numerical values used to generate the plotted data are provided in the Supplementary Data file.

For transparency, the DeepTRACE software is open source and publicly available via GitHub, and archived with Zenodo.

Reporting summary

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

Data availability

Source data for all data-containing figures are provided in the Supplementary Data file. A walkthrough guide, demonstration datasets, and trained models used to reproduce key results are archived at Zenodo[34]. Additional raw data generated during this study are available from the corresponding author upon reasonable request.

Code availability

DeepTRACE v1.0.0, corresponding to this publication, is archived at Zenodo[35]. Following publication, the latest version of DeepTRACE, including future updates will be available via the public GitHub repository (https://github.com/opambos/DeepTRACE). A historical minimal snapshot evaluated during peer review (1st August 2025) corresponding exactly to the walkthrough is archived at Zenodo[36].

Received: 14 August 2025; Accepted: 9 March 2026;

References

1. 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).
2. Elf, J., Li, G.-W. & Xie, X. S. Probing transcription factor dynamics at the single-molecule level in a living cell. *Science* **316**, 1191–1194 (2007).
3. Holden, S. J. et al. High throughput 3D super-resolution microscopy reveals *Caulobacter crescentus* in vivo Z-ring organization. *Proc. Natl. Acad. Sci.* **111**, 4566–4571 (2014).
4. Grimm, J. B. et al. Bright photoactivatable fluorophores for single-molecule imaging. *Nat. Methods* **13**, 985–988 (2016).
5. Hirano, M. et al. A highly photostable and bright green fluorescent protein. *Nat. Biotechnol.* **40**, 1132–1142 (2022).

6. Los, G. V. et al. HaloTag: a novel protein labeling technology for cell imaging and protein analysis. *ACS Chem. Biol.* **3**, 373–382 (2008).
7. Banaz, N., Mäkelä, J. & Uphoff, S. Choosing the right label for single-molecule tracking in live bacteria: side-by-side comparison of photoactivatable fluorescent protein and Halo tag dyes. *J. Phys. Appl. Phys.* **52**, 064002 (2019).
8. Kompa, J. et al. Exchangeable HaloTag ligands for super-resolution fluorescence microscopy. *J. Am. Chem. Soc.* **145**, 3075–3083 (2023).
9. Niederauer, C. et al. Dual-color DNA-PAINT single-particle tracking enables extended studies of membrane protein interactions. *Nat. Commun.* **14**, 4345 (2023).
10. Ries, J. SMAP: a modular super-resolution microscopy analysis platform for SMLM data. *Nat. Methods* **17**, 869–872 (2020).
11. Stracy, M. et al. Live-cell superresolution microscopy reveals the organization of RNA polymerase in the bacterial nucleoid. *Proc. Natl. Acad. Sci.* **112**, E4390–E4399 (2015).
12. Persson, F., Lindén, M., Unoson, C. & Elf, J. Extracting intracellular diffusive states and transition rates from single-molecule tracking data. *Nat. Methods* **10**, 265–269 (2013).
13. Tinevez, J.-Y. et al. TrackMate: An open and extensible platform for single-particle tracking. *Methods* **115**, 80–90 (2017).
14. Hansen, A. S. et al. Robust model-based analysis of single-particle tracking experiments with Spot-On. *eLife* **7**, e33125 (2018).
15. Vink, J. N. A., Brouns, S. J. J. & Hohlbein, J. Extracting transition rates in particle tracking using analytical diffusion distribution analysis. *Biophys. J.* **119**, 1970–1983 (2020).
16. Uphoff, S., Reyes-Lamothe, R., Garza De Leon, F., Sherratt, D. J. & Kapanidis, A. N. Single-molecule DNA repair in live bacteria. *Proc. Natl. Acad. Sci.* **110**, 8063–8068 (2013).
17. Granik, N. et al. Single-particle diffusion characterization by deep learning. *Biophys. J.* **117**, 185–192 (2019).
18. Muñoz-Gil, G. et al. Objective comparison of methods to decode anomalous diffusion. *Nat. Commun.* **12**, 6253 (2021).
19. Muñoz-Gil, G. et al. Quantitative evaluation of methods to analyze motion changes in single-particle experiments. *Nat. Commun.* **16**, 6749 (2025).
20. Kæstel-Hansen, J. et al. Deep learning-assisted analysis of single-particle tracking for automated correlation between diffusion and function. *Nat. Methods* **22**, 1091–1100 (2025).
21. Seckler, H. & Metzler, R. Bayesian deep learning for error estimation in the analysis of anomalous diffusion. *Nat. Commun.* **13**, 6717 (2022).
22. Pinholt, H. D., Bohr, S. S.-R., Iversen, J. F., Boomsma, W. & Hatzakis, N. S. Single-particle diffusional fingerprinting: A machine-learning framework for quantitative analysis of heterogeneous diffusion. *Proc. Natl. Acad. Sci.* **118**, e2104624118 (2021).
23. Andrews, S. S. & Bray, D. Stochastic simulation of chemical reactions with spatial resolution and single molecule detail. *Phys. Biol.* **1**, 137–151 (2004).
24. Lindén, M., Ćurić, V., Boucharin, A., Fange, D. & Elf, J. Simulated single molecule microscopy with SMeagol. *Bioinformatics* **32**, 2394–2395 (2016).
25. Lagage, V., Chen, V. & Uphoff, S. Adaptation delay causes a burst of mutations in bacteria responding to oxidative stress. *EMBO Rep.* **24**, e55640 (2023).
26. Bae, J. & Jeong, H. Segment-based task-specific deep learning framework for detecting state transitions in anomalous diffusion trajectories. *J. Phys. Photonics* **7**, 025027 (2025).
27. Song, D. et al. Deep learning-assisted automated multidimensional single particle tracking in living cells. *Nano Lett.* **24**, 3082–3088 (2024).
28. Kowalek, P., Loch-Olszewska, H., Łaszczuk, Ł, Opała, J. & Szwabiński, J. Boosting the performance of anomalous diffusion classifiers with the proper choice of features. *J. Phys. Math. Theor.* **55**, 244005 (2022).
29. Katz, M. J. & George, E. B. Fractals and the analysis of growth paths. *Bull. Math. Biol.* **47**, 273–286 (1985).
30. Saxton, M. J. Lateral diffusion in an archipelago. Single-particle diffusion. *Biophys. J.* **64**, 1766–1780 (1993).
31. Izeddin, I. et al. Single-molecule tracking in live cells reveals distinct target-search strategies of transcription factors in the nucleus. *eLife* **3**, e02230 (2014).
32. Frankle, J. & Carbin, M. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. *arXiv* https://doi.org/10.48550/arXiv.1803.03635 (2019).
33. Yu, H., Edunov, S., Tian, Y. & Morcos, A. S. Playing the lottery with rewards and multiple languages: lottery tickets in RL and NLP. *arXiv* https://doi.org/10.48550/arXiv.1906.02768 (2020).
34. Pambos, O. J., Wright, J. & Kapanidis, A. DeepTRACE Demonstration Dataset. Zenodo https://doi.org/10.5281/zenodo.16704997 (2025).
35. Pambos, O. J. DeepTRACE v1.0.0. Zenodo https://doi.org/10.5281/zenodo.18869232 (2026).
36. Pambos, O. J. DeepTRACE Reproducibility Snapshot. Zenodo https://doi.org/10.5281/zenodo.16716349 (2025).

Acknowledgements

We thank Valentine Lagage and Stephan Uphoff for providing access to the published DNA polymerase I experimental dataset. We are grateful to Alison Farrar and Mirjam Kummerlin for providing independent human annotations during the development of DeepTRACE. We also thank Mathew Stracy, Isabelle Lorge, Stephan Uphoff, and Sebastian von Hausegger for their feedback on the manuscript. A.N.K. and O.J.P. acknowledge funding from a Wellcome Trust Discovery Award (grant number 26662/Z/22/Z). J.A.R.W. acknowledges funding from the Oxford BBSRC Interdisciplinary Bioscience Doctoral Training Programme.

Author contributions

O.J.P. conceived the DeepTRACE concept and programmed the DeepTRACE software. O.J.P. designed and trained all models. O.J.P. performed the data analysis. The simulation pipeline was co-designed by O.J.P. and J.A.R.W. J.A.R.W. programmed the simulation pipeline, and generated all simulated data. The manuscript and figures were written by O.J.P. with feedback and revisions from A.N.K. The Methods section covering simulations was written by J.A.R.W. with revisions by O.J.P. A.N.K. oversaw the team and acquired funding. All authors approved the manuscript and its submission.

Competing interests

The authors declare the following competing interests: A.N.K. is a co-founder and shareholder of Oxford Nanoimaging Ltd (ONI), a company which manufactures and sells miniaturised single-molecule fluorescence microscopes, including analysis software for single molecule imaging data. The other authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42003-026-09899-y.

Correspondence and requests for materials should be addressed to Oliver J. Pambos.

Peer review information *Communications Biology* thanks Koen J. A. Martens and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Primary Handling Editors: Laura Rodríguez Pérez. A peer review file is available.