TRACKASTRA: Transformer-based cell tracking for live-cell microscopy

Benjamin Gallusser[1] and Martin Weigert[1]

EPFL, Switzerland
{benjamin.gallusser, martin.weigert}@epfl.ch

1 Method

Our proposed TRACKASTRA method operates on raw image sequences and corresponding detections (or segmentation masks) and uses an encoder-decoder transformer to directly predict the pairwise association matrix A between all detections in a local window of consecutive time frames. Specifically, we construct a token for each object and timepoint within the local window and use the sequence of tokens as input to the transformer. The predicted associations $\hat{A}$ are then used as costs in a candidate track graph that is pruned either greedily or via discrete optimization to obtain the final cell tracks. An overview of the full pipeline is shown in Fig. 1. The following sections describe the dataset and training target construction, the transformer architecture, the loss function, the inference and final link assignment, and implementation details.

1.1 Dataset and association matrix construction

Let $I_1, I_2, \ldots, I_T \in \mathbb{R}^{w \times h}$ be an image sequence that is grouped into overlapping windows $S_1, \ldots, S_{T-s} \in \mathbb{R}^{s \times w \times h}$ of size s. Each window S_k contains a set of detections $\{d_i\}$ that each corresponds to a time point $t_i \in \mathbb{N}$, a center point $p_i \in \mathbb{R}^2$, a segmentation mask $m_i \in \{0,1\}^{w \times h}$, and other potential object features $z_i \in \mathbb{R}^k$ such as basic shape descriptors or mean image intensity of the instance. The goal of the model is to predict an association probability matrix $\hat{A} = (\hat{a}_{ij})$ between all d_i in the window S_k. To construct the target association matrix $A = (a_{ij})$ the set of detections $\{d_i\}$ is matched to the set of ground truth objects $V = \{v_k\}$ and their ground truth associations. Each ground truth object again corresponds to a time point t_k, center point p_k, and a segmentation mask m_k and the tracking associations can be described as a directed tree $G = (V, E)$. An edge e_{kl}, $k, l \in V$ exists only if $t_k + 1 = t_l$ and the objects v_k and v_l represent the same cell at different time points, or if v_k is the mother cell of v_l. As a simple matching criterion between detections d_i and ground truth objects v_k we use

$$M_{ik} = \max\left(\mathrm{IoU}(m_i, m_k), 1 - \frac{||p_i - p_k||_2}{\delta_{max}}\right) > 0.5 \quad , \tag{1}$$

where δ_{max} is a distance threshold and IoU denotes the intersection-over-union. The final matching is then obtained by solving a minimum cost bipartite matching problem based on the costs M_{ik} between $\{d_i\}$ and $\{v_k\}$. Finally, for all

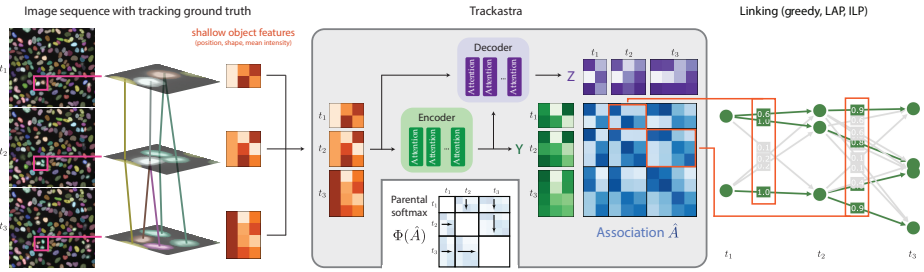

Fig. 1: Overview of TRACKASTRA. Given frame-by-frame object detections in a live-cell video, object features are extracted from a small temporal window and passed as tokens into an encoder-decoder transformer, to predict pairwise associations $\hat{A}$. We apply a *parental softmax* normalisation on $\hat{A}$ to guide the learning directly towards biologically plausible associations. Finally, we build a candidate graph from $\hat{A}$ and prune it with either a greedy algorithm or discrete optimisation to obtain a tracking solution.

matched pairs of detections (d_i, v_{k_i}) and (d_j, v_{k_j}), we set $a_{ij} = 1$ if v_{k_i} and v_{k_j} are part of the same sub-lineage, *i.e.* iff $v_{k_i} \in descendants(v_{k_j})$ or $v_{k_i} \in ancestors(v_{k_j})$, otherwise we set $a_{ij} = 0$. Note that this way, also associations across non-adjacent timepoints as well as appearing and disappearing objects are supported.

1.2 Transformer architecture

The input tokens $x_i \in \mathbb{R}^d$ are constructed by using learned Fourier spatial positional encodings Θ for the detection positions p_i, concatenate them with some features z_i, and projecting them onto the token dimensionality d:

$$x_i = W_{inp}(\Theta(p_i), z_i) \quad . \tag{2}$$

where z_i are the low-dimensional feature vector containing shallow texture and morphological features (such as mask area or mean intensity) and W_{inp} is a linear projection layer mapping the concatenated tensor to $\mathbb{R}^d$. The model consists of an encoder-decoder transformer architecture of $2L$ multi-head attention layers with 4 heads each (*cf.* Fig. 1):

$$\mathcal{A}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}} + M\right)V \tag{3}$$

where Q are the projected attention *queries*, K the projected *keys*, V the projected *values*, and M is a mask disabling attention for all token pairs whose distance is larger than a user defined threshold d_{max}, *i.e.* $M_{ij} = 0$ if $||p_i - p_j||_2 \leq d_{max}$ and $M_{ij} = -\infty$ otherwise. In every attention layer, we use rotary positional embeddings (RoPE [5]) for each intermediate token features according to their corresponding center points p_i to inject positional information. The encoder f transforms the input tokens using L self-attention layers $\mathcal{A}_f^\ell(X, X, X)$

to obtain representations $Y = f(X)$. The decoder g uses L cross-attention layers $\mathcal{A}_g^\ell(X, Y, Y)$ to obtain a second set of representations $Z = g(X, Y)$. In between each attention layers we use a simple two-layer MLP with GeLU activation, layer normalisation and add residual connections following [6]. Finally, we apply two-layer MLPs to Y and Z and compute the logits of the association matrix as their outer product

$$\hat{A} = (\mathrm{MLP}_Y(Y))(\mathrm{MLP}_Z(Z))^T \quad . \tag{4}$$

1.3 Parental softmax

Given the predicted association logits $\hat{A}$ a simple approach to extract association probabilities $\tilde{A} \in (0, 1)$ would be to apply a sigmoid to each entry of $\hat{A}$, *i.e.* $\tilde{A} = \sigma(\hat{A})$. However, this approach does not enforce the combinatorial constraints of cell tracking, *i.e.* the uniqueness of each objects parent while allowing for more than one child, as well as appearance and disappearance of objects. To remedy this, we propose a logit normalisation that we call *parental softmax* and which ensures that the block-wise sum of all entries in the vector of possible parent associations for each d_i is at most one (*cf.* Fig. 1). Concretely, we define the parental softmax $\Phi(\hat{A})$ as

$$\tilde{A} = \Phi(A)_{ij} = \frac{\exp(\hat{A}_{ij})}{1 + \sum_{i' \in \mathcal{P}_j} \exp(\hat{A}_{i'j})} \quad , \tag{5}$$

where $\mathcal{P}_j = \{d_{i'} | t_{i'} = t_j - 1,\ \forall i' \in D\}$ denotes all detections in the frame before detection d_j. Note that adding a constant to the denominator (*quiet softmax*) allows for detections to not be assigned to any parent detection, accommodating for appearing and disappearing objects.

We then define the loss to be minimized during training as

$$\mathcal{L}(A, \hat{A}, W) = \mathcal{L}_{BCE}(A, \Phi(\hat{A}), W) + \lambda \mathcal{L}_{BCE}(A, \sigma(\hat{A}), W) \quad , \tag{6}$$

where $\mathcal{L}_{BCE}$ is the usual element-wise binary cross-entropy loss, $\lambda \in \mathbb{R}$ is a small fixed parameter (we use $\lambda = 10^{-2}$ throughout), and W is a weighting factor for each matrix element. The elementwise weighting terms W are set to

$$w_{ij} = \begin{cases} 0 & t_j - t_i > \Delta t & \textit{temporal cutoff} \\ & \vee\ t_j - t_i < 1 & \textit{only forward links} \\ 1 + \lambda_{\mathrm{div}} & \deg^+(v_{k_i}) = 2 & \textit{dividing cells} \\ 1 + \lambda_{\mathrm{cont}} & \deg^+(v_{k_i}) = 1 & \textit{continuing tracks} \\ 1 & \text{otherwise} & \end{cases} \quad , \tag{7}$$

where $\deg^+(v)$ is the out-degree of vertex v in G. We choose $\Delta t = 2$, $\lambda_{\mathrm{div}} = 10$ and $\lambda_{\mathrm{cont}} = 1$ as fixed hyperparameters. This choice effectively up-weights the loss for cell divisions and continuing tracks, and removes the loss for associations that are not used during the linking step.

1.4 Inference and linking

Inference is done with a sliding window of size s as in training. To obtain global scalar association scores $0 \leq \bar{a}_{i'j'} \leq 1$ from $\tilde{A}^{(1)}, \ldots, \tilde{A}^{(T-s)}$, where i' and j' are global detection indices in a video $I_1, I_2, \ldots, I_T$, we take the mean over the $s-1$ windows that include this association

$$\bar{a}_{i'j'} = \frac{1}{s-1} \sum_{i',j' \in S_t} \hat{a}_{i'j'}^{(S_t)} \quad . \tag{8}$$

Next, we build a candidate graph $G_C = (V, E)$ with a maximum admissible Euclidean distance $dist_{max}$ between detections in adjacent time frames. For this, we use associations $\bar{a}_{i'j'}$ with $t'_j - t'_i = 1$, *i.e.* the upper blockwise diagonal of $\bar{A}$. To generate a first association candidate graph we directly discard small associations with $\bar{a}_{i'j'} < \alpha$ with $\alpha = 0.05$. This candidate graph is then pruned to a solution graph $G_S = (V_S, E_S)$ with $V_S \subseteq V, E_S \subseteq E$ with one of the following linking algorithms:

Greedy We iteratively add edges and their incident nodes to G_S, ordered by descending edge probability, if the edge probability $\theta \geq 0.5$ and if the edge does not violate the biological constraints (*i.e.* at most two children, and at most one parent per vertex)

$$\begin{aligned} \deg^+(v) &\leq 2 \qquad \forall v \in V_s \\ \deg^-(v) &\leq 1 \qquad \forall v \in V_s \quad . \end{aligned} \tag{9}$$

Linear assignment problem (LAP) We use the established two-step LAP as described by Jaqaman *et al.* [3], implemented in [1]. In the first step, linear chains are formed, which are connected to full cell lineages in a second step. We set a maximum linking distance adapted to the respective dataset, and use the default values for all other hyper-parameters.

Integer linear program (ILP) We solve a global ILP with all detections as graph vertices and associations $\{\bar{a}_{i'j'}\}$ with $t_{j'} - t_{i'} = 1$ as edges. The formulation enforces the biological constraints in Eq. (9), as described in [4]. We set the parameters of the ILP, *i.e.* the linear weights of different classes of costs, to values that balance the likelihoods of appearance, disappearance and divisions of cells.

1.5 Implementation details

Training details: We train a single TRACKASTRA model for prototypical 2d and 3d cell tracking datasets on a single GPU (*e.g.* an Nvidia A6000 with 48GB memory). 2d datasets are simply treated as 3d datasets with a single plane in the third dimension. We set window size $s = 4$, embedding dimension $d = 512$, number of encoder and decoder attention layers $L = 5$, and batch size 8.

Shallow object features: As basic object features we use the mean intensity, the object area and the inertia tensor of the object region [2].

Augmentations: We apply the following data augmentations jointly to all frames in a window: flips, arbitrary rotations, shear, scaling, intensity shifting and scaling. Additionally, we apply data augmentations to each frame per window independently: small rotations, shear, translations, additive gaussian noise.

References

1. Fukai, Y.T., Kawaguchi, K.: LapTrack: linear assignment particle tracking with tunable metrics. Bioinformatics **39**(1) (2023). https://doi.org/10.1093/bioinformatics/btac799 4
2. Jähne, B.: Spatio-temporal image processing: theory and scientific applications. chap. 8: Tensor Methods. Springer (1993) 5
3. Jaqaman, K., Loerke, D., Mettlen, M., Kuwata, H., Grinstein, S., Schmid, S.L., Danuser, G.: Robust single-particle tracking in live-cell time-lapse sequences. Nature Methods **5**(8), 695–702 (2008). https://doi.org/10.1038/nmeth.1237 4
4. Malin-Mayor, C., Hirsch, P., Guignard, L., McDole, K., Wan, Y., Lemon, W.C., Kainmueller, D., Keller, P.J., Preibisch, S., Funke, J.: Automated reconstruction of whole-embryo cell lineages by learning from sparse annotations. Nature Biotechnology **41**(1), 44–49 (2023). https://doi.org/10.1038/s41587-022-01427-7 4
5. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. Neurocomputing **568**, 127063 (2024) 2
6. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017) 3