Tracking by Instance Detection in Fluorescence Microscopy: A Meta-Learning Approach for Nuclear Cell Tracking

Roman Frátrik[1], Elena Šikudová[1]

[1]Charles University, Department of Software and Computer Science Education, Malostranské nám. 25 118 00 Praha 1, Czech Republic

Abstract

In this work, we adapt the tracking-by-instance-detection method, originally proposed for natural image sequences, to fluorescence microscopy, focusing on the `Fluo-N2DH-SIM+` dataset. We apply model-agnostic meta-learning (MAML) to initialise object detectors for rapid domain adaptation to individual cell instances, without requiring architectural modifications. We compare two detector families under this framework: an anchor-based approach (RetinaNet) and an anchor-free approach (FCOS), and investigate the effect of meta-model initialisation when trained on large general-purpose visual datasets against cell-specific microscopy data.

Keywords

cell tracking, meta-learning, MAML, object detection, fluorescence microscopy, tracking by detection, domain adaptation

1. Introduction

Quantitative analysis of live-cell fluorescence microscopy increasingly depends on reliable, automated tracking of individual cells across time. The *Cell Tracking Challenge* [1, 2] has established standard datasets and metrics for this task, but most top-performing methods are specialised pipelines that require substantial amounts of densely annotated training data, a resource that is expensive to obtain for every new microscope, cell line, or staining protocol.

In the natural-image domain, *tracking by detection* has become a dominant paradigm: an object detector is run on each frame, and the resulting detections are linked across time [3]. Wang et al. [4] recently reframed single-object tracking itself as a specialised detection problem, *tracking by instance detection*: a generic object detector is meta-trained so that, given a single example of a target in the first frame, it can be converted into an instance-specific detector with only a few gradient steps. This is achieved with model-agnostic meta-learning (MAML) [5], which learns an initialisation that enables fast adaptation to a new task (here, a new target instance). The approach is architecture-agnostic and does not require structural modification of the underlying detector.

Problem statement. Tracking a single nucleus through a fluorescence sequence is difficult for several reasons. First, nuclei are dim and weakly textured, so the appearance cues that identify one instance are limited (**low signal-to-noise ratio**). Second, mitosis splits one lineage into two, and a single-instance tracker cannot follow both daughters at once (**cell division**). Third, nuclei are densely packed and look almost identical, so a tracker is easily drawn onto a neighbour (**crowding and occlusion**). Finally, dense per-frame masks are costly to produce for every new microscope or cell line, which limits supervised training (**scarce annotations**).

Tracking via instance detection is a natural fit for this setting: annotated targets are scarce, yet a single seed annotation per cell is inexpensive, and the method requires no architectural changes. In

ITAT'26: Information Technologies – Applications and Theory, September 25–29, 2026, Biele Karpaty, Slovakia
roman.fratrik974@student.cuni.cz (R. Frátrik); sikudova@cgg.mff.cuni.cz (E. Šikudová)
0009-0004-2728-3837 (R. Frátrik); 0000-0003-4572-4064 (E. Šikudová)
© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-11-09

CEUR-WS.org/Vol-4323/paper6.pdf

this paper, we ask whether the paradigm transfers from natural video to fluorescence microscopy, and which design decisions matter most.

Study design. This work is organised around four experimental components. First, we adapt the MAML-based "detector as tracker" of Wang et al. [4] to nuclear cell tracking on simulated `Fluo-N2DH-SIM+` videos [6, 7] (**tracking by instance detection for microscopy**). Second, we compare an anchor-based RetinaNet head [8] against an anchor-free FCOS head [9], each meta-initialised on either MS-COCO [10] or LIVECell [11], under an otherwise identical protocol (**a 2×2 study of head and pre-training domain**). Third, we test all four configurations on the same videos using per-video Wilcoxon signed-rank tests with Holm correction [12], reporting a lineage-aware tracking accuracy (TRA) alongside our other lineage metrics (**paired, effect-size-aware evaluation**). Finally, we inspect the raw probability map each head emits to explain *why* the configurations differ (**score-map analysis of the heads**).

We find that neither factor wins on its own: the head and the meta-training domain *interact*. The anchor-free FCOS head pairs best with the in-domain LIVECell initialisation, whereas the anchor-based RetinaNet head pairs best with the general-purpose MS-COCO initialisation; these matched pairings clearly outperform the mismatched ones.

2. Related Work

Cell tracking. Classical cell-tracking pipelines combine a segmentation or detection stage with a linking stage based on overlap, nearest-neighbour, or graph optimisation [1, 2]. Such methods can be highly accurate but typically require dense, frame-by-frame annotations for training and careful per-dataset tuning. Because real annotated sequences are limited, simulation has become an important tool: CytoPacq and its underlying phantom generator synthesise realistic multi-dimensional cell-image sequences with exact ground truth, including the `Fluo-N2DH-SIM+` nuclei used here [6, 7].

Tracking by detection and instance detection. Tracking by detection links per-frame detections into trajectories [3]. Siamese trackers instead learn a similarity function between a template and a search region [13]. Tracking by instance detection [4] unifies these views: it keeps a standard detector but specialises it online to a single target instance, leveraging the detector's strong localisation prior whilst adapting its classification branch to the specific object.

3. Method

Tracking a single nucleus through a video of dividing, visually near-identical cells is a question of identity: at every frame, the tracker must decide whether a candidate region is *this* cell and not one of its many similar-looking, possibly newly-divided neighbours. A detector trained once and applied uniformly to all cells has no notion of which cell is the target; it can find nuclei in general, but cannot commit to tracking any particular one. We address this by meta-training a detector that can be rapidly specialised, from only the target's first-frame box, into an instance-specific detector for that one cell, turning a generic detection problem into a per-target one. The remainder of this section details the tracking-by-instance-detection pipeline and its MAML-based meta-initialisation; the detector architecture and the two detection heads we compare; the two domains used to meta-train them; and the online procedure that adapts and tracks at test time.

3.1. Tracking by instance detection

Figure 1 summarises the three phases of the method. The detector is first meta-trained *offline* into an initialisation (θ, α) (Section 3.2); at test time, this initialisation is adapted to each target cell from

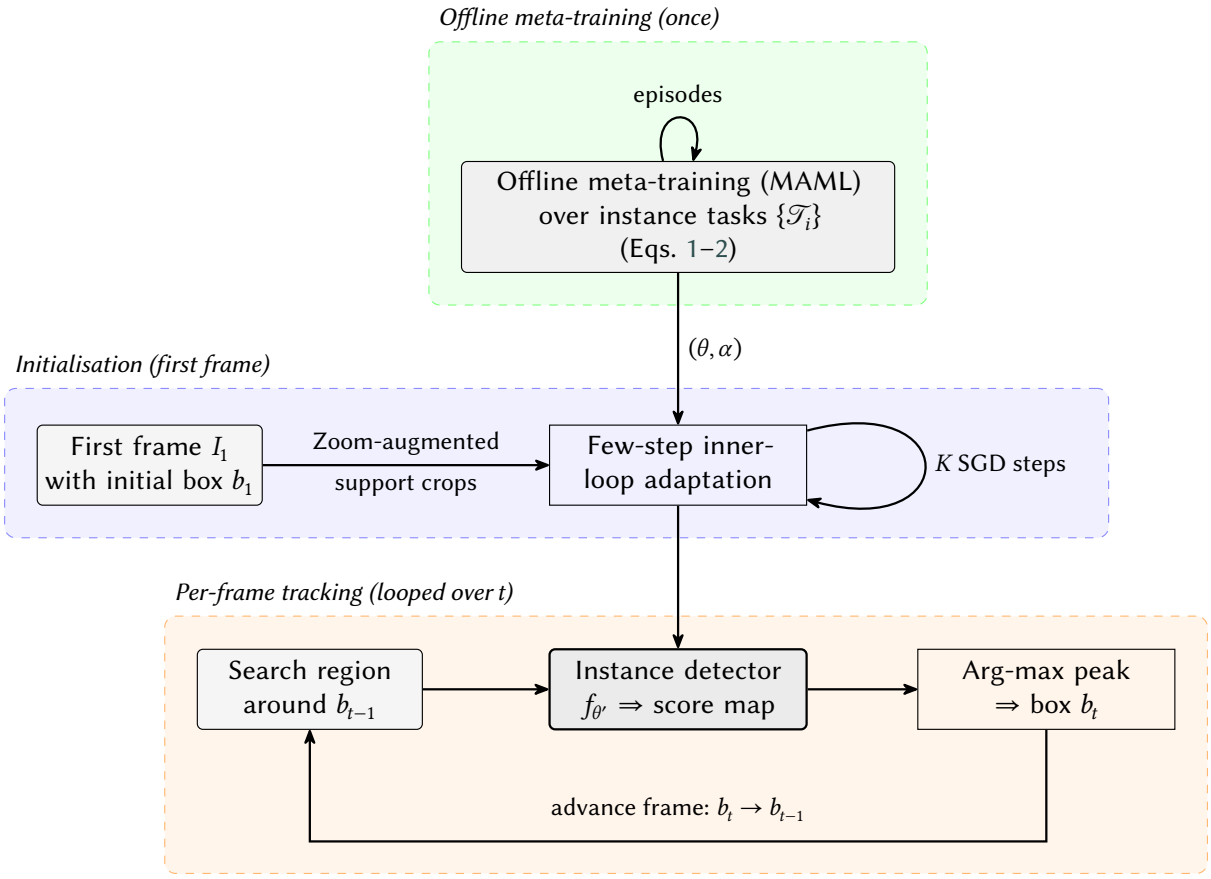


Figure 1: Tracking by instance detection adapted to nuclear cell tracking. The detector is first meta-trained *offline* with MAML over many instance tasks to obtain the initialisation (θ, α) (Eqs. (1)–(2); Section 3.2). Given a video sequence of frames $\{I_t\}_{t=1}^{T}$, from the first frame I_1 and the initial box b_1 of a target nucleus (taken from the ground-truth mask of frame 0) we build a small support set of zoom-augmented crops of the target and run a few inner-loop gradient steps (Eq. (1)) from the meta-initialised parameters (θ, α), yielding an instance-specific detector $f_{\theta'}$ whose positive class is the single target instance. For each subsequent frame we apply $f_{\theta'}$ within a search region centred on the previous estimate b_{t-1} and take the highest-scoring detection as b_t, propagating the track frame to frame so that a single detection failure terminates it; this makes the task demanding and the evaluation discriminative. One independent tracker is spawned per nucleus present in the first frame.

its first-frame box, and the resulting instance detector is then applied frame by frame. We adopt this tracking-by-instance-detection pipeline from Wang et al. [4] and apply it to nuclear cell tracking.

3.2. Meta-initialisation with MAML

Let $\mathcal{T}_i$ denote an instance-adaptation task with support loss $L_{\mathcal{T}_i}$. MAML [5] seeks an initialisation θ from which a few inner gradient steps yield task-adapted parameters. Starting from $\theta_0 = \theta$, the k-th inner step is

$$\theta_k = \theta_{k-1} - \alpha \odot \nabla_\theta L_{\mathcal{T}_i}^{\text{sup}}(f_{\theta_{k-1}}), \tag{1}$$

where, following Wang et al. [4], the step size α is not a scalar but a *learnable per-channel learning rate* (one value per convolution-kernel output channel) that is meta-trained jointly with θ, and $\odot$ denotes element-wise multiplication. The meta-objective minimises the post-adaptation loss on a *target* view of the same instance, accumulated over the whole inner trajectory $\{\theta_k\}_{k=0}^{K}$, with meta learning rate β and trajectory weights γ_k that anneal linearly over training epochs from uniform towards a schedule that up-weights the later inner-loop, with the final, fully adapted step θ_K weighted highest.

$$(\theta, \alpha) \leftarrow (\theta, \alpha) - \beta \nabla_{(\theta,\alpha)} \sum_{\mathcal{T}_i} \sum_{k=0}^{K} \gamma_k \, L_{\mathcal{T}_i}^{\text{tgt}}(f_{\theta_k}), \tag{2}$$

Only the adaptable parameters receive gradients (Section 3.3); the frozen backbone does not. During inference, Eq. (1) is the only adaptation performed per target, so the quality of the initialisation (θ, α) directly determines tracking performance. We obtain (θ, α) in two ways (Section 3.4) and keep the meta-learning procedure otherwise fixed.

3.3. Detector and detection heads

We instantiate the detector f_θ on a ResNet-18 [14] backbone that is partitioned into a *frozen* part and an *adaptable* part. The stem and residual blocks 1-3 are pre-trained on ImageNet and kept frozen (batch-norm in evaluation mode) throughout both meta-training and tracking. The final residual block is duplicated into two independent copies, one feeding a classification branch and one a regression branch. Concretely, we make two separate deep copies of that block, both initialised from the shared ImageNet weights and thereafter, each copy receives its own gradients and adapts independently. Both consume the same frozen feature map produced by the earlier blocks, and together with the detection head, they form the adaptable parameters updated by Eq. (1). The detector operates on a single feature scale of a fixed-size input crop, so no feature pyramid is used.

We compare two heads. The **RetinaNet head** [8] is anchor-based: it performs dense anchor classification with sigmoid focal loss ($\alpha_\mathrm{f} = 0.25$, $\gamma_\mathrm{f} = 2$) and smooth-L_1 box regression at positive anchors. The **FCOS head** [9] is anchor-free: it predicts per location a centre-ness score and *ltrb* box offsets, using a positive-weighted mean-squared-error loss on the centre-ness map and a smooth-L_1 loss on the decoded boxes; the centre-ness map doubles as the detection score map. Both heads are trained and adapted using the same MAML algorithm.

3.4. Meta-training domains

We meta-initialise each head on two domains. **MS-COCO** [10] is a large general-purpose dataset of natural images and the domain used in the original method [4]. **LIVECell** [11] is a large dataset of label-free microscopy images of cultured cells, placing it in-domain with respect to our microscopy target. We test the intuition that in-domain pre-training (LIVECell) should transfer better than general-purpose pre-training.

3.5. Online tracking and adaptation

At test time, the meta-initialised detector is turned into a tracker in two adaptation phases that reuse the *same* inner loop of Eq. (1). On the first frame, we run a heavier *domain adaptation*: five inner steps on a support set of zoom-augmented crops of the initial box. In each subsequent frame, we crop a search region around the previous estimate (1.5× the target box), resize it to the network's input size, and run a single forward pass to obtain the score and box maps. The score map is then multiplied by a SiamRPN-style shape penalty that discourages abrupt changes in size or aspect, and blended with a cosine (Hanning) window that discourages large centre jumps; the peak of the result gives the new box, which is interpolation-smoothed against the previous one. A light *online update* runs every ten frames, and additionally whenever the peak-to-sidelobe ratio signals a competing distractor. It is a single inner-loop step (Eq. (1)) on a small, capped memory of *self-labelled* exemplars: whenever a detection is confident (score ≥ 0.8), its crop and predicted box are appended to a support set that always retains the first-frame crop and is capped at 30 entries. The step minimises the *same* detection loss used in meta-training (focal loss for the RetinaNet head, centre-ness loss for FCOS) and updates only the classification branch, since adapting the FCOS exponential regression head online is numerically unstable; the regression branch is frozen throughout tracking. The search region also keeps the target near the crop centre to limit interference from the surrounding nuclei.

Domain-matched input. The two meta-training domains differ in appearance, so we feed each head the input representation that matches the domain it was initialised on. The MS-COCO initialised heads

receive the raw fluorescence frame, min–max mapped to 8-bit and replicated to three channels as a generic natural image. The LIVECell-initialised heads instead receive the same frame after LIVECell-style background correction, so that their low-contrast, label-free appearance matches the microscopy images on which the LIVECell model was meta-trained. We call this per-head choice *domain-matched input*; feeding both domains a single shared preprocessing degrades whichever head is mismatched, so matching the input to the initialisation is what makes the four configurations comparable.

Fixed search scale. The search region is always *centred* on the previous estimate, so it follows the target's motion; the two options concern only its *size*. Under the *fixed* scale the search-region size is anchored to the initial object and held constant, whereas under the *free* scale it follows the size of the most recent predicted box. The free scale creates a feedback loop: an overly large predicted box enlarges the next search region, which then admits even larger boxes, so the window can inflate until it spans the whole frame. This size drift, not positional drift, is what the fixed scale removes; fixing the size does not impair position tracking, because the region still recentres on the target each frame. We therefore report the fixed scale, which leaves the detection head and the meta-initialisation domain as the only factors varied across configurations; the free scale is left as a tuning option for future work.

4. Experimental Setup

This section describes the experimental methodology: the dataset and its generation process, the lineage-aware metrics used to score tracking performance, and the statistical tests used to compare configurations. Throughout, we denote each configuration by ⟨head⟩-⟨domain⟩, abbreviating RetinaNet as Retina (so Retina-COCO is the RetinaNet head meta-trained on MS-COCO).

4.1. Dataset

We use simulated nuclear-cell videos of the `Fluo-N2DH-SIM+` type, generated with CytoPacq [6, 7]. Each video contains 100 frames, and the number of visible nuclei increases over time as cells divide, so later frames are substantially more crowded than early ones. Simulation provides exact per-frame ground-truth masks and lineages, thereby removing annotation noise from the evaluation and allowing us to isolate the tracker's behaviour. The detectors are meta-trained only on MS-COCO or LIVECell and evaluated on these simulated videos, so the two stages use disjoint data.

4.2. Evaluation metrics: lineage-aware (non-split) scoring

Each tracker is initialised on a single cell (the founder, F) in the first frame. The tracker's task is to follow F and all cells descended from it through mitosis. This entire set is called the *lineage* of F, denoted as $\mathscr{L}(F)$. In each frame t, the tracker's predicted bounding box for F ($b_t(F)$) is compared to all ground-truth cells in $\mathscr{L}(F)$ present in that frame using the Intersection-over-Union (IoU) metric

$$\mathrm{IoU}(A, B) = \frac{|A \cap B|}{|A \cup B|}. \tag{3}$$

The score for frame t ($s_t(F)$) is the highest IoU between the predicted bounding box and the bounding boxes of any $\mathscr{L}(F)$ cell present in the frame. Mathematically

$$s_t(F) = \max_{c \in \mathscr{L}(F) \cap G_t} \mathrm{IoU}\big(b_t(F), b_t(c)\big), \tag{4}$$

where G_t is the set of ground-truth cells in frame t.

The frame is labelled a true positive (TP) when $s_t(F) \geq \tau$ (we use $\tau = 0.5$); a false negative (FN) when the lineage is present but no box reaches τ (including frames where the target is lost, $s_t = 0$); and a false positive (FP) when a box is assigned to the incorrect lineage.

This evaluation does not require the tracker to follow both daughter cells after a mitosis event. Instead, the tracker only needs to stay on any cell in $\mathscr{L}(F)$. This is because the current tracking method does not explicitly model cell division. Thus, the evaluation answers the question "Is the tracker still on this lineage?" rather than "Did it follow the correct daughter?".

We report three per-video quantities. **Mean IoU** is the mean of $s_t(F)$ over all frames in which the lineage is present, counting a lost target as zero; it measures localisation quality and persistence jointly. **Success rate** (lineage recall) is the fraction of present-lineage frames that are true positives, TP/(TP + FN), and captures how often the lineage is held. **Precision**, TP/(TP + FP), is the fraction of emitted boxes that are correct.

Tracking accuracy (TRA). As a fourth, graph-based summary, we report a lineage-aware tracking accuracy

$$\mathrm{TRA} = 1 - \frac{\min(\mathrm{AOGM}, \mathrm{AOGM}_0)}{\mathrm{AOGM}_0}, \tag{5}$$

the normalised Acyclic Oriented Graph Matching (AOGM) error, where AOGM_0 is the cost of building the reference track graph from scratch (every ground-truth vertex an FN, every edge missing). We compute it on the *same* non-split, lineage-collapsed graph used above: each present-lineage frame is a vertex, scored TP/FP/FN by the rule just given, and the only edges are the temporal links along a founder's presence chain, so a broken link counts as an association error. Because cell divisions are collapsed, there are no split vertices and no parent edges. This is therefore a *non-split* tracking accuracy and *not* the standard split-aware Cell-Tracking-Challenge TRA, although it uses the same graph-edit weights.

4.3. Statistical methodology

Each configuration is evaluated on the *same* videos, so all comparisons are paired, with one per-video value as the unit of observation (per-frame IoUs are temporally correlated and are not treated as independent). In the two-factor (head × domain) design, we compare each pair with the exact paired Wilcoxon signed-rank test [15], using a *two-sided* alternative. A two-sided test is appropriate here because we make no directional hypothesis about which head or meta-training domain should win, and because the two factors need not act in the same direction: they, in fact, form a crossover (Section 5), for which no single one-sided hypothesis is well defined. Two-sided testing is also the more conservative choice. We Holm-correct the p-values across the entire family of 8 comparisons (two metrics × four pairwise tests), correcting them jointly rather than per metric [16, 12], and report the rank-biserial correlation of matched pairs, $r \in [-1, 1]$, as the effect size.

5. Results

This section reports the tracking performance of the four detector configurations, two detection heads (FCOS, RetinaNet) crossed with two meta-initialisation domains (MS-COCO, LIVECell) evaluated on the 25 simulated videos described in (Section 4). We begin with aggregate per-video metrics (Table 1), then examine the raw score maps each head produces, pooled over all founders and frames (Section 5.1), and close with paired statistical tests that assess whether the observed differences are reliable (Section 5.2).

The high-level picture is that no single configuration dominates across all metrics. The two *matched* pairings FCOS with LIVECell and RetinaNet with COCO achieve the best success rate, and TRA are practically tied (TRA 0.277 vs 0.276); both *mismatched* pairings trail them. Mean IoU tells a slightly different story: the three configurations other than FCOS-COCO cluster closely (0.33–0.35), while FCOS-COCO alone is clearly weaker for this metric. Precision is moderate across the board (0.84–0.94), consistent with low inter-lineage movement. The two orderings appear to conflict, since Retina-LIVECell has the highest mean IoU yet the lowest success rate, but they measure different things. Success rate thresholds each frame at IoU ≥ 0.5, whereas mean IoU averages the raw overlap and counts a lost

Table 1
Mean per-video tracking metrics over the simulated videos. Higher is better for all columns; best value per column in bold. TRA is our lineage-aware, non-split tracking accuracy (Section 4.2), not the standard split-aware Cell-Tracking-Challenge score. "Head" is the detection paradigm; "Init." is the meta-initialisation domain.

Head	Init.	Mean IoU	Success rate	Precision	TRA
FCOS (anchor-free)	MS-COCO	0.237	0.214	0.842	0.230
FCOS (anchor-free)	LIVECell	0.330	**0.273**	**0.935**	**0.277**
RetinaNet (anchor-based)	MS-COCO	0.343	0.260	0.843	0.276
RetinaNet (anchor-based)	LIVECell	**0.352**	0.151	0.881	0.159

frame as zero. Of the four heads, Retina-LIVECell stays on the correct cell most persistently (highest peak-in-GT, 0.82, and lowest localisation error; Table 2) but only loosely: its boxes fit at a mean overlap of about 0.34 (its pooled score-map IoU, 0.338, almost equals its per-video mean IoU, 0.352), which lies below the 0.5 threshold, so they rarely cross it (lowest success rate) yet rarely collapse to zero (highest mean IoU). FCOS-COCO is the opposite: it loses the target outright most often (lowest peak-in-GT, 0.41), and those true zeros pull its mean IoU down to 0.237, but on the frames it does hold the cell, its boxes fit tightly enough to clear 0.5 more often than Retina-LIVECell's. Mean IoU, therefore, rewards loose persistence while success rate rewards threshold-crossing localisation, and the two can diverge.

5.1. Detection head analysis

To look beyond the aggregate tracking numbers, we examine the raw score map each head emits within its search region, pooled over *every* founder and *every* frame of the 25 videos ($n = 23{,}863$ founder-frames; Table 2), each variant on the domain-matched input of Section 3.5. Because the descriptors are collected *during live tracking*, they measure localisation as actually achieved. Once a tracker drifts, its search crop no longer contains the target, which lowers the peak-in-GT and increases the localisation error. By combining per-frame localisation and target persistence, the descriptors are directly comparable to the tracking metrics of Table 1.

Three observations follow. First, localisation is *not* uniformly solved: the peak lands inside a ground-truth box on 41% of frames for FCOS-COCO but 82% for Retina-LIVECell. Second, the pooled map IoU (0.24–0.34) tracks the per-video mean IoU of Table 1 (0.24–0.35) almost exactly, confirming that the score-map and per-video analyses measure the same quantity from two directions. Third, and most telling, the raw-map ranking does *not* predict tracking success: Retina-LIVECell produces the best-centred map of all four variants (highest peak-in-GT 0.82, lowest localisation error) yet has the *lowest* success rate and TRA (Table 1), because its peak sits on the cell but its boxes are too loosely sized to clear the $\tau = 0.5$ threshold. The *peak* column must be read with the same caution: a high pooled peak is not itself a sign of sharper localisation. FCOS-COCO's centre-ness is well calibrated, near zero (≈ 0.04) on the off-cell crops it sees after drift and about 0.66 when on the cell, so its *pooled* peak (0.28) is dominated by those honest off-cell values; the LIVECell-trained centre-ness instead *saturates*, firing ≈ 0.8 even off-cell, which inflates its pooled peak (0.87) without improving box fit. Figure 2 illustrates the on-cell case qualitatively for one well-tracked founder, where all four heads concentrate their map on the nucleus.

Figure 2, therefore, complements Table 2 rather than repeating it: the table reports behaviour pooled over all founders and frames, whereas the figure shows, for one well-tracked lineage, what an on-cell map looks like when tracking succeeds. The primary evidence for the head × domain interaction remains the per-video analysis of Tables 1–3.

5.2. Statistical significance

The above descriptive ranking is supported by paired tests (Table 3). Regarding the success rate, all four paired comparisons are significant after Holm correction and form a crossover: FCOS prefers LIVECell,

Table 2
Score-map quality pooled over all founders and every frame of the 25 videos (n = 23,863 founder-frames), each variant evaluated on domain-matched input (Section 3.5). Statistics are measured during live tracking, so a drifted tracker's off-cell crop lowers peak-in-GT and raises the localisation error; a high peak may also reflect centre-ness saturation rather than sharper localisation (see text). A sharper, better-localised map has higher PSR, lower entropy, higher peak-in-GT, lower localisation error (in cell diagonals), and higher map IoU.

Variant	Peak	PSR	Entropy	Peak-in-GT	Loc. err	IoU
FCOS-COCO	0.279	7.91	0.886	0.407	2.187	0.241
FCOS-LIVECell	0.874	3.64	0.743	0.699	0.587	0.320
Retina-COCO	0.559	3.67	0.955	0.656	0.530	0.316
Retina-LIVECell	0.490	6.38	0.805	0.821	0.395	0.338

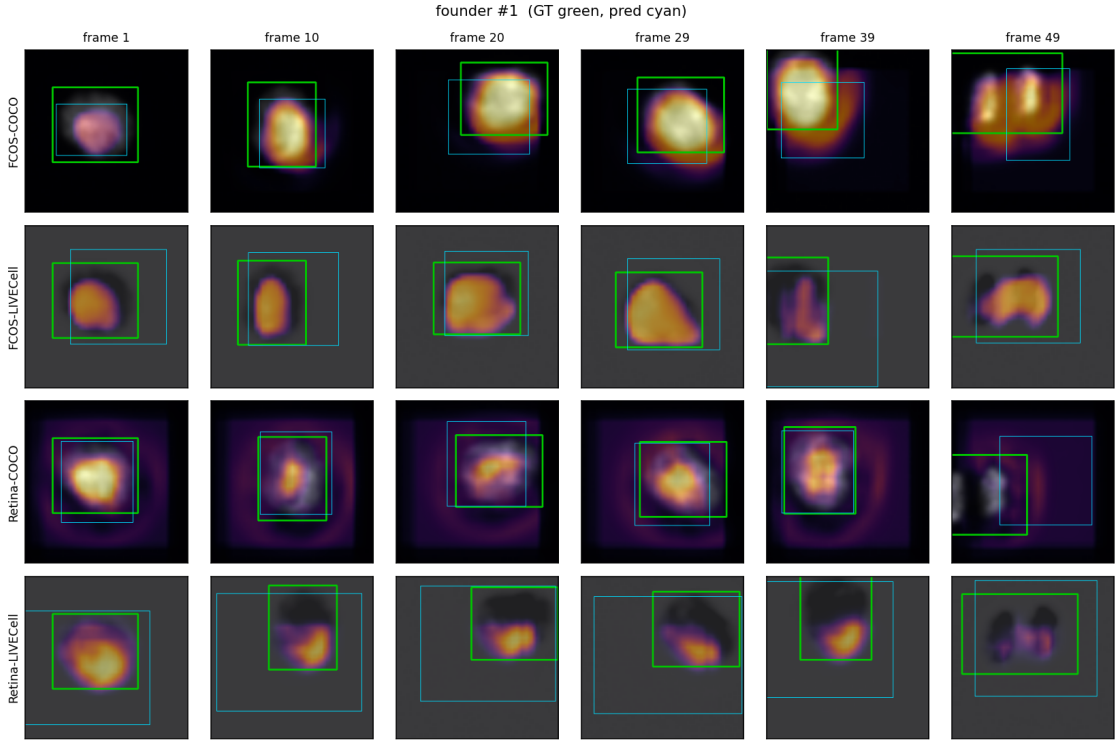


Figure 2: Score maps for one example lineage (founder #1) at six frames of the clip, one row per variant up to 50 frames for visualisation purposes. Each tile shows the search crop with the head's probability map overlaid (higher scores are brighter), and the ground-truth (green) and predicted (cyan) boxes drawn. With domain-matched input, every variant concentrates its map on the nucleus and keeps the predicted box on the cell; Retina-COCO shows the tightest green/cyan agreement for this founder. This lineage is a best-case illustration in which every head stays on the cell for the whole clip; pooled over all founders (Table 2), the heads are far less uniform.

whilst RetinaNet prefers COCO. For mean IoU, the significant effects isolate only FCOS-COCO as the weakest configuration, beaten by both Retina-COCO and FCOS-LIVECell; the other differences are not significant.

5.3. Summary of findings

Together, the results support the following five conclusions.

No single configuration is best on all metrics: FCOS-LIVECell leads in success rate, TRA, and precision, whilst Retina-LIVECell achieves the highest mean IoU (Table 1).

Table 3
Paired Wilcoxon signed-rank comparisons over $n = 25$ videos. "median Δ" is the median paired difference for the first-named configuration minus the second (a positive value favours the first); r is the matched-pairs rank-biserial effect size; p_{Holm} is Holm-corrected across all 8 comparisons. A bullet (•) marks significance at family-wise $\alpha = 0.05$.

Metric	Comparison	median Δ	r	p_{Holm}	sig.
Mean IoU	FCOS vs. RetinaNet (COCO)	−0.110	−1.00	4.8×10^{-7}	•
	FCOS vs. RetinaNet (LIVECell)	−0.020	−0.48	0.068	
	LIVECell vs. COCO (FCOS head)	+0.087	+0.98	1.8×10^{-6}	•
	LIVECell vs. COCO (Retina head)	+0.019	+0.32	0.17	
Success rate	FCOS vs. RetinaNet (COCO)	−0.048	−0.60	0.022	•
	FCOS vs. RetinaNet (LIVECell)	+0.112	+0.99	8.4×10^{-7}	•
	LIVECell vs. COCO (FCOS head)	+0.066	+0.74	2.5×10^{-3}	•
	LIVECell vs. COCO (Retina head)	−0.102	−0.95	7.4×10^{-6}	•

The head and pre-training domain interact, with FCOS preferring LIVECell and RetinaNet preferring COCO; all four paired success-rate comparisons are significant and form a crossover (Table 3).

The matched pairings dominate in tracking quality: FCOS-LIVECell and Retina-COCO achieve the highest success rate and TRA, practically tied and ahead of both mismatched pairings.

The mean IoU is a misleading summary statistic: Retina-LIVECell has the highest mean IoU but the lowest success rate and TRA.

Finally, raw score-map quality does not by itself explain tracking: pooled over all founders and frames, the best-localised map (Retina-LIVECell) belongs to the *worst*-tracking configuration, because its boxes are too loosely sized to clear the overlap threshold (Table 2).

An anomaly deserves mention. For Retina-COCO, activation is elevated around the edges of the search region (Figure 2), which is possibly an artefact of anchor tiling. We leave a full diagnosis to future work.

6. Discussion

Our main finding is an interaction rather than a main effect: neither the head nor the pre-training domain is better on its own. FCOS pairs best with LIVECell and RetinaNet with COCO, so an ablation that fixed one factor could have named either head or either domain the "winner," depending on the other; for meta-learned trackers, the two should be chosen jointly.

Score-map analysis (Section 5.1) shows that raw localisation and tracking success are only loosely coupled: pooled over all founders and frames, the configuration with the best-centred map (Retina-LIVECell) tracks *worst*, its boxes too loose to clear the overlap threshold, whereas for the others the decisive factor is robustness to distractors. "In-domain" pre-training is thus not universally better, since LIVECell helps FCOS but, in terms of success rate, hurts RetinaNet relative to COCO.

Limitations. The study has two main limitations. First, we evaluate on simulated `Fluo-N2DH-SIM+` videos; although the simulator models realistic image formation, the results should be confirmed on real microscopy sequences (**simulated data**). Second, the tracker follows a single instance and cannot follow both daughters after a division, so we report the lineage-aware non-split metric (Section 4.2), which measures lineage *persistence* rather than full *reconstruction* (**no mitosis modelling**).

7. Conclusion

We adapted tracking via instance detection to fluorescence microscopy nuclear cell tracking, using MAML to convert a generic detector into an instance-specific tracker without architectural changes.

Across simulated videos and a paired statistical analysis, we found no universally best configuration: the detection head and the meta-training domain interact, so that the anchor-free FCOS head pairs best with the in-domain LIVECell initialisation and the anchor-based RetinaNet head with the general-purpose MS-COCO one, the two matched pairings leading in success rate and TRA. A score-map analysis pooled over all founders and frames showed that raw localisation and tracking success are only loosely coupled: the best-localised configuration was not the best-tracking one.

7.1. Future work

Several directions follow from this study. The most immediate is explicit mitosis modelling (**cell splitting**): detecting a mitotic event for a tracked nucleus and spawning a second instance tracker so that both daughters are followed, producing a complete lineage tree rather than a single maintained branch. Only a tracker that follows every daughter can be scored fairly under the stricter, split-aware AOGM/TRA measure [1, 2]: our implementation already computes that measure, but applied to the present single-instance tracker, it would penalise every unfollowed daughter, so we report the non-split variant here and defer the split-aware evaluation to that setting. Beyond mitosis, the findings should be confirmed on real fluorescence sequences rather than only on the simulator (**validation on real microscopy**). Finally, the fixed search scale and online-update schedule, both held constant here, warrant a dedicated sensitivity study (**search-scale and adaptation tuning**).

Acknowledgments

The authors thank the organisers of the Cell Tracking Challenge and the developers of the CytoPacq simulation service for the datasets and tools used in this study. CytoPacq web service [6] (https://cbia.fi.muni.cz/simulator).

Declaration on Generative AI

During the preparation of this work, the authors used Claude OPUS for formatting assistance, citation management, and grammar and spelling checks. After using these tools, the authors reviewed and edited content as needed and take full responsibility for the publication's content.

References

[1] M. Maška, V. Ulman, D. Svoboda, P. Matula, P. Matula, C. Ederra, A. Urbiola, T. E. na, S. Venkatesan, D. M. W. Balak, et al., A benchmark for comparison of cell tracking algorithms, Bioinformatics 30 (2014) 1609–1617. doi:10.1093/bioinformatics/btu080.

[2] V. Ulman, M. Maška, K. E. G. Magnusson, O. Ronneberger, C. Haubold, N. Harder, P. Matula, P. Matula, D. Svoboda, M. Radojevic, et al., An objective comparison of cell-tracking algorithms, Nature Methods 14 (2017) 1141–1152. doi:10.1038/nmeth.4473.

[3] A. Bewley, Z. Ge, L. Ott, F. Ramos, B. Upcroft, Simple online and realtime tracking, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), 2016, pp. 3464–3468. doi:10.1109/ICIP.2016.7533003.

[4] G. Wang, C. Luo, X. Sun, Z. Xiong, W. Zeng, Tracking by instance detection: A meta-learning approach, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 6288–6297. doi:10.1109/CVPR42600.2020.00632.

[5] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: Proceedings of the 34th International Conference on Machine Learning (ICML), volume 70 of *Proceedings of Machine Learning Research*, 2017, pp. 1126–1135.

[6] D. Wiesner, D. Svoboda, M. Maška, M. Kozubek, CytoPacq: a web-interface for simulating multi-dimensional cell imaging, Bioinformatics 35 (2019) 4531–4533. doi:10.1093/bioinformatics/btz417.

[7] D. Svoboda, M. Kozubek, S. Stejskal, Generation of digital phantoms of cell nuclei and simulation of image formation in 3-d image cytometry, Cytometry Part A 75A (2009) 494–509. doi:10.1002/cyto.a.20714.

[8] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988. doi:10.1109/ICCV.2017.324.

[9] Z. Tian, C. Shen, H. Chen, T. He, FCOS: Fully convolutional one-stage object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 9627–9636. doi:10.1109/ICCV.2019.00972.

[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: Common objects in context, in: Computer Vision – ECCV 2014, volume 8693 of *Lecture Notes in Computer Science*, 2014, pp. 740–755. doi:10.1007/978-3-319-10602-1_48.

[11] C. Edlund, T. R. Jackson, N. Khalid, N. Bevan, T. Dale, A. Dengel, S. Ahmed, J. Trygg, R. Sjögren, LIVECell—a large-scale dataset for label-free live cell segmentation, Nature Methods 18 (2021) 1038–1045. doi:10.1038/s41592-021-01249-6.

[12] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.

[13] L. Bertinetto, J. Valmadre, J. ao F. Henriques, A. Vedaldi, P. H. S. Torr, Fully-convolutional siamese networks for object tracking, in: Computer Vision – ECCV 2016 Workshops, volume 9914 of *Lecture Notes in Computer Science*, 2016, pp. 850–865. doi:10.1007/978-3-319-48881-3_56.

[14] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778. doi:10.1109/CVPR.2016.90.

[15] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics Bulletin 1 (1945) 80–83. doi:10.2307/3001968.

[16] S. Holm, A simple sequentially rejective multiple test procedure, Scandinavian Journal of Statistics 6 (1979) 65–70.

Implementation details

All four reported configurations (FCOS/RetinaNet head × MS-COCO/LIVECell init) share all settings listed here; they differ only in the head architecture and the meta-initialisation domain. Rows marked † are post-processing knobs carried over from the base method and are *not* tuned for cell motion; tuning is left for future work.

Component	Parameter	Value
Backbone	Architecture	ResNet-18 (ImageNet pre-trained)
	Frozen layers	Stem, blocks 1–3
	Adaptable layers	Block 4 + head
	Input resolution	263×263, single feature scale
Meta-training	Episode structure	1 instance; 3 support crops (zoom 1.08×), 1 target
	Inner loop	4 steps; learnable per-channel step (base 10^{-3})
	Outer loop	Adam, lr $= 10^{-4}$; grad. clip $= 10$
	Gradient order	First-order (early epochs) → second-order
Schedule	Epochs / episodes	20 epochs × 10,000 episodes
	Batch size	64
	Loss	γ-annealed trajectory loss
	Hardware	Single NVIDIA RTX 3080 GPU
Tracking	First-frame adapt	5 inner steps
	Online update	1 step / 10 frames (classification branch only)
	Regression branch	Frozen during tracking
	Window influence†	0.4
	Shape penalty†	$k = 0.04$