ZERO-SHOT SAM AND SAMURAI WITH CROSS-BACKBONE SPLICE AND CROP-WINDOW CONSENSUS FOR HYPERSPECTRAL OBJECT TRACKING

Tze-Hsiang Tang

Independent Researcher, Taiwan
seantangth@gmail.com

ABSTRACT

We describe the winning entry of the 2026 Hyperspectral Object Tracking Challenge. Neither of its trackers is trained on hyperspectral data: two frozen video segmentation models, SAM 3 and SAM 2.1 driven by SAMURAI, run zero-shot on the false-colour videos supplied with the dataset. Three components sit on top. A two-pass target-aware crop re-tracks small targets inside a window derived from the first-pass trajectory; three such windows are tracked and fused per frame by a medoid rule. A cross-backbone splice takes the secondary tracker's box wherever the primary one has frozen. A ridge regressor that predicts the IoU difference between the two trackers arbitrates the remaining frames. The entry ranked 7th on the public leaderboard and first in the final evaluation, run by the organisers on unseen videos. We also report what did not help, including three attempts to use the hyperspectral data beyond false colour.

***Index Terms*—** Hyperspectral object tracking, foundation models, zero-shot tracking, Segment Anything, tracker fusion

1. INTRODUCTION

A hyperspectral video records a spectrum at every pixel, and the spectrum depends on the material of the surface. Tracking in such videos promises robustness where colour and shape are ambiguous [1]. Most hyperspectral trackers adapt a tracker built for colour video to the extra bands: they select or group bands, fuse the responses of several band groups, or add spectral branches to a pretrained backbone [2, 3, 4, 5, 6, 7].

The 2026 Hyperspectral Object Tracking Challenge [8] provides videos from three snapshot mosaic cameras (VIS, 16 bands; NIR, 25 bands; RedNIR, 15 bands), each with a spatially aligned false-colour rendering. There are 406 annotated training videos and 75 validation videos whose annotations are withheld. Teams were first ranked on the validation videos through a public leaderboard (Ranking A; 256 teams entered the challenge). The top teams then handed in code, which the organisers ran on unseen videos to produce the final standing (Ranking B).

This paper describes the entry that ranked 7th in Ranking A and first in Ranking B. It started from a plain question: how far does a frozen, general-purpose video segmentation model go on this task when it sees only the false-colour frames? The answer frames everything that follows. SAM 3 [9], prompted with the first-frame box and nothing else, already scores 0.674 success AUC on Ranking A. None of our attempts to improve on it by fine-tuning, or by feeding it spectral bands instead of false colour, helped (Sec. 3.4). What did help was engineering around the two ways in which the frozen model fails: it switches to a similar object when the scene is cluttered, and it stops producing a mask when the target is lost. We see the result as a baseline that spectral methods should be measured against, not as evidence that the spectrum is useless; Sec. 4 returns to this point.

The contributions are: (i) a single-variable comparison of zero-shot backbones on the challenge server; (ii) a two-pass target-aware crop whose gain, as a comparison with a larger input size shows, is not explained by resolution and which we attribute mainly to the removal of distractors, together with a crop-window consensus that makes it robust to the choice of window; (iii) a cross-backbone splice and a regression selector motivated by the evaluation metric; and (iv) a list of negative results. Code: `https://github.com/seantangth/hotc2026-ranking-b`.

2. METHOD

Fig. 1 gives the pipeline. A box is $b = (x, y, w, h)$; A_t and B_t denote the boxes of the primary and the secondary leg at frame t, and G_t the ground truth.

2.1. Zero-shot tracker legs

The pipeline is zero-shot at the tracker level: both legs run with their released weights, are prompted with the first-frame box only, and see only the false-colour frames. The mask of each frame is converted to its tight bounding box. If the mask is empty the previous box is repeated. We count consecutive identical boxes, $r_1 = 0$ and $r_t = r_{t-1} + 1$ if $b_t = b_{t-1}$, else $r_t = 0$, and use this *frozen-run* length as a label-free proxy for tracking failure. The primary leg is SAM 3 [9] at its native input size of 1008×1008. The secondary leg is SAM 2.1 (Hiera-L) [10] with the motion-aware memory selection of

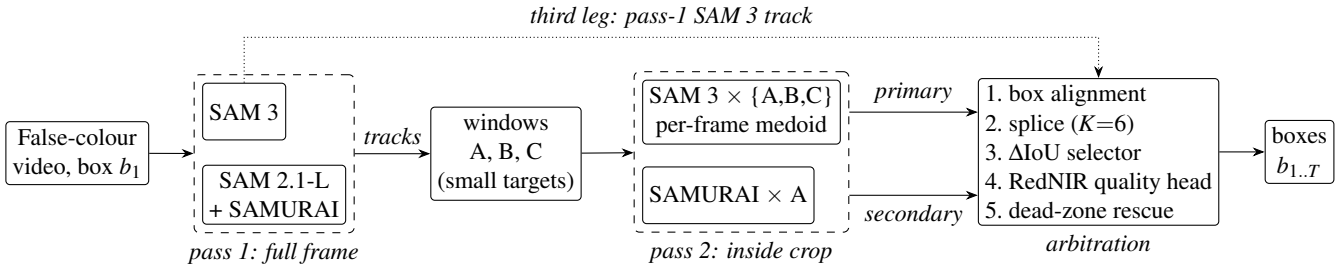


Fig. 1. Pipeline. Both trackers are frozen and prompted only with the first-frame box. Pass 2 is run for sequences with small targets; all other sequences keep their pass-1 tracks.

SAMURAI [11], whose Kalman state we reset at the start of every sequence.

2.2. Two-pass target-aware crop

Targets are small (the median annotated box in the training videos is 28×36 pixels, and the smallest are a few pixels wide), and the scenes often contain several objects of the same kind. Pass 1 tracks every sequence on the full frame with both legs. For a sequence whose first-frame target scale $s = \sqrt{w_1 h_1}$ is below 32 pixels, the union of all pass-1 boxes of both legs is enlarged by $\max(2s, 32)$ pixels on each side and clipped to the frame, which gives one fixed window for the sequence. If the window covers less than 55% of the frame, the frames are cropped to it, both legs track again inside the crop at a higher effective resolution (pass 2), and the boxes are mapped back; otherwise the pass-1 track is kept. Using both trajectories matters: a leg that has lost the target is frozen in the wrong place, and its own envelope would cut the target out. The window is constant over time, so the memory of the tracker sees a stable view.

Crop-window consensus. Tracking inside a crop is recurrent, and we found it sensitive to the window: a small change of the window can make the track diverge (Sec. 3.3). SAM 3 is therefore tracked under three windows computed from the same pass-1 output: (A) the window above; (B) the sequence split into two halves, each with its own, tighter window, the second half initialised from the pass-1 box; (C) the envelope of SAM 3's own trajectory only. Each variant applies the area test on its own and keeps the pass-1 track where it fails. Per frame we keep the *medoid* of the three boxes, a median-of-three consensus that returns the box with the largest summed IoU to the other two, with ties going to A. No box is synthesised, and with three candidates the rule discards a run that has drifted to another object.

2.3. Cross-backbone splice

Let A be the consensus SAM 3 track and B the SAMURAI track (window A). The splice, a frozen-run rescue across backbones, outputs $S_t = B_t$ whenever $r_t^A \geq K$ and $r_t^B < 2$, and $S_t = A_t$ otherwise, with $K = 6$. The rule never has to judge which of two live boxes is better, which is the hard problem; it fires only where the primary leg has most likely stopped producing masks. SAM 2.1 is the weaker tracker overall, but on such frames its box is far better than a frozen one: by 0.42 IoU per replaced frame in a validation probe on full-frame tracks with $K = 24$.

2.4. Learned arbitration

The success AUC averages the success rate over 50 IoU thresholds and is, up to this discretisation, the mean per-frame IoU. Under this surrogate, replacing the spliced output S_t by B_t changes the score by d_t/N, where $d_t = \mathrm{IoU}(B_t, G_t) - \mathrm{IoU}(S_t, G_t)$ and N is the number of frames, and the ideal rule takes B_t iff $\mathbb{E}[d_t \mid \mathbf{f}_t] > 0$. This motivates regressing the gain instead of classifying which box is better: a classifier gives a near tie and a rescue worth 0.4 IoU the same loss, and near ties dominate the data. We fit a ridge regressor ($\lambda = 1$, standardised features) on the training videos to predict d_t, excluding ties and using sample weights $\min(|d_t|, 0.5)$ that concentrate the fit on frames where the decision matters. Its 25-dimensional input $\mathbf{f}_t$ (22 geometric terms and a one-hot code of the three cameras) needs no image access and is computed from the corrected boxes of the two legs: $\mathrm{IoU}(A_t, B_t)$; the log-areas and their ratio; the aspect ratios; the centre distance relative to the target size; the frame-to-frame change of both boxes; both frozen-run lengths; the relative position in the sequence; and the box sizes. B_t is taken when the prediction is at least $\tau = 0.05$ and B is not frozen; the score is flat around this value (Sec. 3.3). The splice can be seen as a hand-written special case for frames where $|d_t|$ is large and its sign is obvious.

Two small rules complete the chain. A *RedNIR quality head*, a logistic model on the same features fitted on a 65-video split of the training set to predict which leg has the higher IoU, is applied only to RedNIR videos, the camera on which the pipeline is weakest, and only where the legs disagree strongly ($\mathrm{IoU}(A_t, B_t) < 0.3$). A *dead-zone rescue* handles frames where $r_t^A \geq K$ and B is frozen too ($r_t^B \geq 2$): the pass-1 full-frame SAM 3 box is used if that run is alive.

Table 1. Backbone × crop on Ranking A. The crop here is the first version of Sec. 2.2: one window per sequence, area threshold 40% (21 of 75 sequences, 26% of the frames), identical windows for both backbones.

Backbone	Full frame	+ crop	Δ
SAM 2.1-L + SAMURAI	0.66608	0.67199	+0.00591
SAM 3	0.67383	0.69128	+0.01745
Δ	+0.00775	+0.01929	

The first-frame box is always returned as given. On Ranking A the splice fires on 513 of 26,860 frames, the selector on 7,652, the RedNIR head on 127 and the rescue on 37. These counts include frames where the two boxes coincide; the final output differs from the corrected primary track on 6,086 frames (22.7%).

2.5. Coordinate-convention correction

Masks yield tight boxes. As a coordinate-convention correction, the top and left edges of every predicted box are moved outward by one pixel ($x-1$, $y-1$, $w+1$, $h+1$, clipped at the image border). It is applied to all legs before the arbitration, and the given first-frame box is kept as it is.

3. EXPERIMENTS

3.1. Setup

Unless stated otherwise, numbers are success AUC [12] on the 75 Ranking A videos (40 VIS, 22 NIR, 13 RedNIR; 26,860 frames), computed by the challenge server with the frames of all videos pooled. Design choices were validated with submissions to this server that changed a single variable; Ranking B, run by the organisers on unseen videos, is the independent test. In the released pipeline the training annotations are used only to fit the two arbitration heads of Sec. 2.4. The pipeline runs in PyTorch with bfloat16 autocast on one NVIDIA A10 and needs 4–5 hours for 75 videos (two full-frame passes and four crop passes). With pinned package versions it is deterministic: a second run on another machine with the same GPU model reproduced the submission row for row.

3.2. Backbone and crop

Table 1 isolates the two largest effects. The crop is worth three times more on SAM 3 than on SAM 2.1, and the interaction (+0.0115 over an additive model) exceeds both main effects. Among zero-shot backbones, model size mattered within the SAM 2.1 family (base-plus 0.61940, large 0.66608), and a distractor-aware memory [13] lowered the score of the large model to 0.65611.

Table 2. The released pipeline on Ranking A (server AUC). Every row is regenerated end to end from the videos by the public code.

Configuration	AUC	Δ
Single pass: both legs, correction, splice	0.68148	
+ two-pass crop and learned arbitration	0.70701	+0.02553
+ crop-window consensus (submitted)	0.71096	+0.00395

Is the crop gain a matter of resolution? The crop affects 26% of the frames at a median area magnification of 4.2, which raises the patch budget available to the target, weighted over frames, by a factor of roughly 1.83. Raising the SAM 3 input size from 1008 to 1344 for all frames buys nearly the same budget (1.78) but only +0.0024, against +0.0175 for the crop; adding the crop on top of the larger input is 0.0208 worse than the crop at the native size. The gain is therefore not explained by resolution; we attribute most of it to removing distractors from the field of view, which matches the dominant failure we observed for SAM 3, a stable switch to a similar object. Relaxing the area threshold gives a plateau (0.40: 0.69128; 0.50: 0.69234; 0.55: 0.69292; 0.60: 0.69276; 0.70: 0.69111); we use 0.55.

3.3. Ablation

Table 2 gives the released pipeline at three levels. The single-pass variant runs both legs on the full frame and applies the coordinate correction and the splice, with no crop and no learned part. The two-pass crop with the learned arbitration adds 0.0255, and the crop-window consensus a further 0.0040; 0.71096 is the score of the submitted method.

Submissions that changed a single variable give the contribution of the individual parts on the two-pass chain: the coordinate correction +0.0142, the splice +0.0060, the ΔIoU selector +0.0023, the RedNIR quality head +0.0009 and the dead-zone rescue +0.0003; the crop is isolated in Table 1. The contributions are not additive, as the interaction in Table 1 already shows. The splice and the selector sit on plateaus rather than peaks. The score varies by less than 0.0001 for K between 5 and 8 and drops by 0.0008 at $K = 2$ and by 0.0011 at $K = 24$; it varies by 0.0002 for τ between 0 and 0.05 and drops by 0.0009 at $\tau = 0.15$. Replacing the secondary backbone by other SAM 2.1 variants moved the splice result by less than 0.0004.

The consensus answers a sensitivity of the crop pass. Between two runs of the crop pass whose windows differed only slightly, most of the score difference sat in five cropped sequences, although in four of them the full-frame tracks of the two runs agreed at IoU 0.93–0.99: the tracker had not changed, the recurrent tracking inside the crop had diverged. Instead of relying on one window, the pipeline tracks under the three windows of Sec. 2.2, all computed from the run's

Table 3. Interventions that did not help (Δ server AUC on Ranking A against the stated reference, except the last row).

Intervention	Reference	Δ
Linear spectral adapter (NIR)	SAMURAI-L	−0.0273
Fisher-projected bands as input	full pipeline	+0.0002
Pan-sharpened crops	SAM 3 + crop	−0.0275
Input size 1344	SAM 3	+0.0024
Input size 1344 + crop	SAM 3 + crop	−0.0208
Four temporal crop segments	SAM 3 + crop	−0.0139
Distractor-aware memory [13]	SAMURAI-L	−0.0100
Memory stride 5	SAM 3	−0.0202
Fine-tuned decoder replaces SAM 3	full pipeline	−0.0349
LoRAT-L / LoRAT-g [14]‡	SAM 3	−0.049

‡ zero-shot; local 65-video split of the training set.

own pass 1, and takes the per-frame medoid.

3.4. What did not help

Table 3 lists what we tested and dropped. Three of the interventions use the hyperspectral data beyond the false-colour rendering. A linear adapter that maps the NIR bands to three channels, initialised by a least-squares fit to the false-colour image and trained on the NIR training videos, lost 0.027. A probe on 21 training videos, using the first-frame box only, showed that the full spectrum does separate the target from its surround better than false colour in NIR and RedNIR: the median gain in the held-out AUC of a Fisher discriminant is +0.032 and +0.034, against −0.001 in VIS. Yet replacing the input channels of SAM 3 by the corresponding Fisher projection, computed from the first-frame box only, changed the score by +0.0002. Pan-sharpening the crops with the luminance of the raw mosaic lost 0.028.

Fine-tuning did not help either. With the mask decoder of SAM 3 (4.2M parameters) fine-tuned on the training videos and used in place of the frozen model, the server score dropped by 0.035, through target switches on cluttered sequences. LoRAT [14], a one-stream transformer trained for single-object tracking, scored 0.049 below SAM 3 zero-shot on a local split of the training videos, and its 1.1B-parameter variant differed from the 0.3B one by 0.0002. The obstacle for such trackers on false-colour hyperspectral video appears to be the domain, not capacity.

3.5. Final evaluation

Of 256 teams, the top teams of Ranking A were invited to submit code. Nine entries were evaluated by the organisers, who ran the submitted code on unseen videos, and ours ranked first [15].

4. DISCUSSION AND LIMITATIONS

The pipeline uses no spectral information beyond the three-channel false-colour projection. Our negative results do not show that the spectrum is uninformative: the separability probe says otherwise for NIR and RedNIR. They show that pushing spectral bands through the three input channels of a frozen RGB model, or training a light adapter in front of it, does not turn that separability into tracking accuracy. Whether an adapter trained with a temporally consistent objective can do so is open, and the numbers above give it a baseline to beat.

The pipeline is slow (about two frames per second end to end) and is not meant for real-time use. Recurrent tracking inside a crop remains sensitive to the window; the consensus reduces the dependence on one window but does not remove it.

5. CONCLUSION

Frozen SAM-family trackers on false-colour video, combined with a distractor-removing crop, a consensus over crop windows and a metric-motivated arbitration between two backbones, won the 2026 Hyperspectral Object Tracking Challenge without fine-tuning either tracker and without using the spectral bands. The components exploit two label-free signals, frozen runs and disagreement between backbones. We hope the negative results are as useful to the community as the positive ones.

6. ACKNOWLEDGMENT

The author thanks the organisers of the challenge for the dataset and for running the final evaluation.

7. REFERENCES

[1] Fengchao Xiong, Jun Zhou, and Yuntao Qian, "Material based object tracking in hyperspectral videos," *IEEE Transactions on Image Processing*, vol. 29, pp. 3719–3733, 2020.

[2] Zhuanfeng Li, Fengchao Xiong, Jun Zhou, Jing Wang, Jianfeng Lu, and Yuntao Qian, "BAE-Net: A band attention aware ensemble network for hyperspectral object tracking," in *IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 2106–2110.

[3] Zhuanfeng Li, Fengchao Xiong, Jun Zhou, Jianfeng Lu, and Yuntao Qian, "Learning a deep ensemble network with band importance for hyperspectral object tracking," *IEEE Transactions on Image Processing*, vol. 32, pp. 2901–2914, 2023.

[4] Zhenqi Liu, Xinyu Wang, Yanfei Zhong, Meng Shu, and Chen Sun, "SiamHYPER: Learning a hyperspectral object tracker from an RGB-based tracker," *IEEE Transactions on Image Processing*, vol. 31, pp. 7116–7129, 2022.

[5] Yuzeng Chen, Qiangqiang Yuan, Yuqi Tang, Yi Xiao, Jiang He, and Liangpei Zhang, "SPIRIT: Spectral awareness interaction network with dynamic template for hyperspectral object tracking," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–16, 2024.

[6] Rafał Muszyński and Hiep Luong, "HELIOS: Hyperspectral hindsight OSTracker," in *13th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, 2023, pp. 1–5.

[7] Shaheer Mohamed, Tharindu Fernando, Sridha Sridharan, Peyman Moghadam, and Clinton Fookes, "Spectral-enhanced transformers: Leveraging large-scale pretrained models for hyperspectral object tracking," in *14th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, 2024, pp. 1–5.

[8] HotTracking2026, "Hyperspectral object tracking challenge 2026," `https://kaggle.com/competitions/hyperspectral-object-tracking-challenge-2026`, 2026, Kaggle.

[9] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, et al., "SAM 3: Segment anything with concepts," *arXiv preprint arXiv:2511.16719*, 2025.

[10] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, et al., "SAM 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024.

[11] Cheng-Yen Yang, Hsiang-Wei Huang, Wenhao Chai, Zhongyu Jiang, and Jenq-Neng Hwang, "SAMURAI: Adapting segment anything model for zero-shot visual tracking with motion-aware memory," *arXiv preprint arXiv:2411.11922*, 2024.

[12] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Online object tracking: A benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 2411–2418.

[13] Jovana Videnovic, Alan Lukezic, and Matej Kristan, "A distractor-aware memory for visual object tracking with SAM2," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 24255–24264.

[14] Liting Lin, Heng Fan, Zhipeng Zhang, Yaowei Wang, Yong Xu, and Haibin Ling, "Tracking meets LoRA: Faster training, larger model, stronger performance," in *European Conference on Computer Vision (ECCV)*, 2024.

[15] HotTracking2026, "Final competition result notice," `https://www.kaggle.com/competitions/hyperspectral-object-tracking-challenge-2026/discussion/741628`, 2026, Kaggle discussion.