
Where to Hitch? Finding Small, Rare Parts Across Long Spatiotemporal Contexts

Cheng-Lin Hsieh^{1*} Zilin Wang^{1*} Stella X. Yu¹
Enrique Corona² Sandeep Sasidharan² Erol Dogan Sumer²

¹University of Michigan ²Hyundai America Technical Center

{clhsieh,zilinwan,stellayu}@umich.edu {ecorona,ssasidharan,esumer}@hatci.com

*Equal contribution

Abstract

Foundation models struggle to find a small, rare, and visually plain part in a long, high-resolution video. Their fixed input resolution makes processing every frame in full detail costly, while downsizing hides the part. Naming the part does not single it out either: the concept is rare, and its appearance alone is not distinctive. We study this problem for hitch assist, which locates a trailer’s coupler as the vehicle backs toward it. The coupler covers a median 0.2% of the image, and text-prompted models often box a larger structure. We introduce **HitchIt**, a training-free method that selects the spatial and temporal context a frozen video foundation model sees. HitchIt specifies the part with boxed examples from other videos and searches a few late views, where the coupler is large. In each view, it narrows the search from the trailer through its tongue to the coupler: the trailer and tongue are easier to find, and each crop enlarges the coupler in the model’s input and excludes distractors. It then tracks the best box backward and forward through the video. On 20 densely annotated videos, HitchIt reaches 0.627 mean box overlap (mIoU), versus at most 0.067 for text-prompted models, including SAM3 on the same late views. On 67 separate videos, part-by-part search raises mIoU from 0.513 to 0.645 over full-image search. We will release our dense coupler annotations.

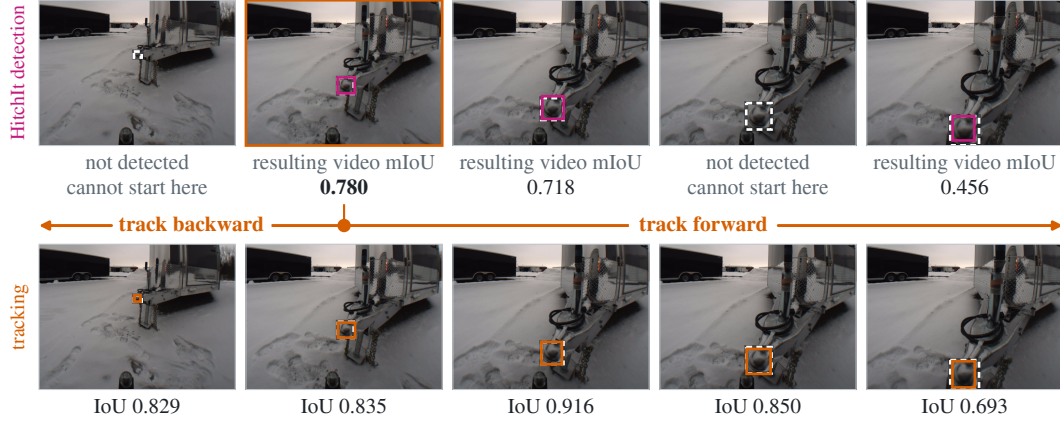
1 Introduction

Hitch-assist systems must locate a trailer’s coupler as the vehicle backs toward it [1, 2]. The coupler is the socket at the tip of the trailer’s tongue that receives the vehicle’s hitch ball (Fig. 2a). It is hard to find for three reasons. It is small: in a rear-camera video, it is tiny at first and grows as the vehicle approaches (Fig. 1a). It is rare: a specialized towing part, uncommon in everyday images. It is visually plain: similar metal surfaces appear on many objects, including nearby jacks, chains, other trailers, and the vehicle’s hitch ball, so appearance alone does not distinguish it.

The coupler’s size makes its localization a long-context problem in both space and time. A recording spans many high-resolution frames, but a foundation model sees each input at a fixed resolution. Processing every frame in full detail is costly, and downsizing the frames hides the part. The coupler’s rarity and plain appearance add a second difficulty: its name does not identify it. Text-prompted foundation models such as Grounding DINO [3], OWLv2 [4], and SAM3 [5] often return a box around a larger structure, such as the trailer, instead of the coupler.

We separate identifying the part from following it, and the key is identifying it correctly in the right frame. Given one correct box in a clear view, a tracker follows the coupler back to early views where it is only 21 pixels across (Fig. 2b). Which frame supplies that box matters: in Fig. 1a, some views yield no detection, and tracking from the highlighted detection gives 0.780 video mIoU, versus 0.456 when starting from the fixed final view. Identification therefore does not need the whole video. It

(a) Starting-view choice affects final tracking accuracy.



(b) HitchIt compared with existing models.

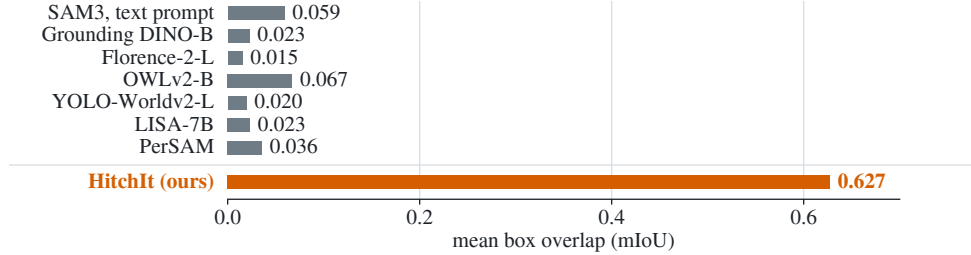


Figure 1: **Finding a suitable starting view improves the resulting track.** (a) Top: HitchIt’s detection, with retrieved examples and part-by-part search, runs on each view separately. A detected box starts the same tracker, and the resulting video mIoU is measured on the same video. A view without a detection cannot start a track. Bottom: boxes tracked from the highlighted detection, with per-frame IoU. Magenta: HitchIt detections; orange: tracked boxes; dashed white: annotations. (b) Published baselines [3–9] and HitchIt on 20 videos; SAM3 shows its text-prompt setting.

needs the right spatial and temporal context: a clear view where the part is large, and a search region that gives it enough of the model’s input and excludes distractors.

We introduce HitchIt, a training-free method that obtains this identification automatically with a frozen video foundation model (Fig. 3). HitchIt retrieves boxed examples of the part from a small bank drawn from other videos, choosing them by the part’s appearance. It shows each example beside the search image as one input, so the example’s box specifies what to find without marking where it is. The search proceeds from the trailer to its tongue to the coupler. The trailer and tongue are larger and more conspicuous than the coupler, and each crop around them gives the coupler more of the model’s input and leaves out distractors elsewhere. HitchIt runs this search on four late views, where the coupler appears largest, and tracks the best box backward and forward through the video.

We evaluate HitchIt on 20 densely annotated approach videos and on 67 further videos with one labeled frame each. On the dense videos, HitchIt localizes the coupler far more accurately than text-prompted foundation models (Fig. 1b). With the same frozen model, a name prompt fails even on the late views where boxed examples succeed, so the coupler’s size alone does not explain the difficulty. Identification succeeds mainly in late views, where the coupler is large. Part-by-part search outperforms fixed image tiles, and on the 67 separate videos it clearly improves on searching the full image. These results support a simple division of work for long videos: spend the model’s input on a few well-chosen views and regions to identify the target, then track it through the recording.

Our contributions are as follows. (1) **Selective context for hard-to-find parts:** HitchIt identifies a small, rare, and visually plain part by choosing which views, image regions, and examples a frozen foundation model sees, without training or a target box in the new video. (2) **Controlled evidence:** holding the search or the tracker fixed, we measure how view choice, example retrieval,

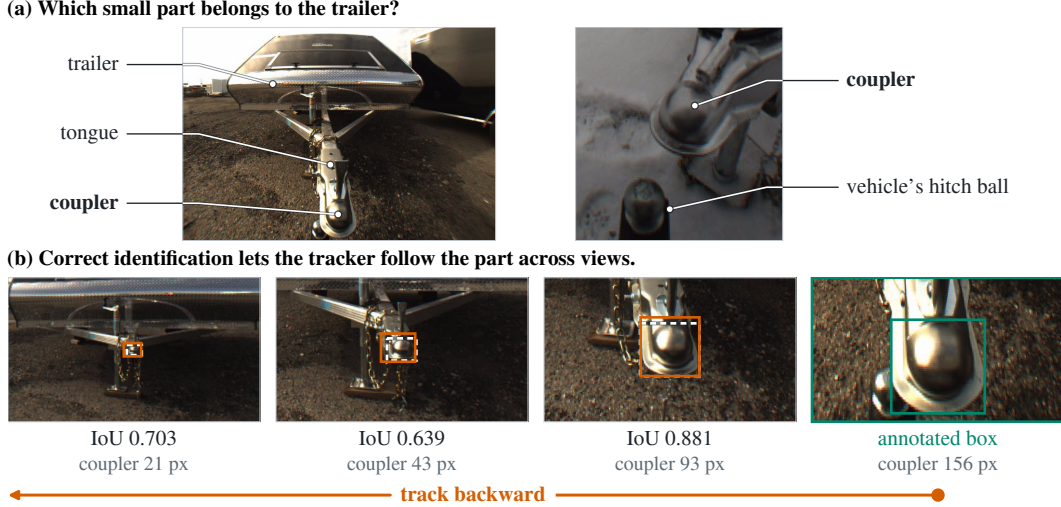


Figure 2: **The coupler is difficult to identify, but a correct starting box supports tracking.** (a) The coupler sits at the tip of the trailer’s tongue; the nearby hitch ball belongs to the vehicle. (b) The tracker receives the annotated coupler box in the rightmost view (green) and predicts boxes in earlier views (orange). Dashed white: annotations used only for evaluation. Close-ups share one display scale; coupler size is $\sqrt{\text{box area}}$ in original pixels. This oracle demonstration isolates propagation from detection: the challenge is identifying a suitable frame and detecting the coupler in it.

and part-by-part search each affect localization. (3) **Dense coupler annotations:** we annotate coupler boxes throughout 20 approach videos from the MSU Trailer Database [1] and will release them.

2 Related Work

Long videos and memory-based tracking. Long-video benchmarks test whether models can use information spread across a recording [10–12]. Memory-based segmenters follow a target through a video once it is marked in one frame [5, 13–15]. HitchIt uses such a tracker unchanged and addresses the step before it: finding the target automatically in a few chosen views. Selective labeling picks samples for human annotation [16, 17]; HitchIt picks views for automatic localization.

Text and visual prompts. Open-vocabulary detectors and segmenters localize regions named by text [3, 4, 6–8]. Visual prompting instead specifies the target with examples [5, 18–20]. Training-free methods segment new images from annotated references using frozen features [9, 21, 22]. HitchIt also uses examples without training. It retrieves them by part appearance from a bank of other videos and places each beside the search image, so the model’s own exemplar prompt works across images.

Guided visual search. V* [23] lets a multimodal language model search a high-resolution image for small details by zooming into likely regions. HitchIt also zooms selectively, but its search path follows the target’s physical parts, from the trailer to its tongue to the coupler.

Coupler perception. Prior hitch-assist systems train detectors on task-specific labels [1, 2]. The MSU Trailer Database provides approach videos with coupler centers and close-up contours [1]. We study training-free 2D localization and add dense coupler boxes for 20 of its videos.

3 HitchIt

Task. HitchIt receives an unannotated approach video and a bank of boxed examples of the trailer, tongue, and coupler from other videos. It outputs a coupler box in every frame. Because HitchIt processes recorded videos offline, a clear late view can help localize the coupler in earlier frames.

(a) Use a boxed example to find the part in a new image.



(b) Find the small target through its larger parts.



(c) Search four late views, then track the most confident box.

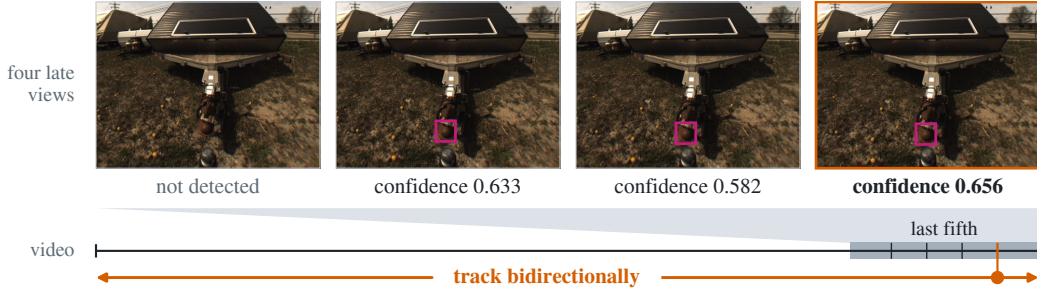


Figure 3: **HitchIt chooses the examples, image regions, and views that the model sees.** (a) A boxed example from another video is paired with the unmarked search image, here at the coupler step; the prediction comes from that pair. (b) Predicted trailer and tongue boxes narrow the search to the coupler, each step using (a); white outlines mark the region searched at the next step. (c) The search in (b) runs on four views from the last fifth of the video, and the most confident box is tracked to both ends. Labels give detection confidence, not tracking accuracy. Magenta: detections; green: annotated examples; orange: tracking. These choices focus the search on a clear view of the part.

Overview. HitchIt uses a frozen video foundation model, SAM3 [5], with two abilities. Given a box around an example object, the model finds matching objects in the same input image. Given a box in one frame, it tracks the object through the video. The model resizes every input to a fixed resolution, which limits the detail it sees. HitchIt decides how to use this fixed input (Fig. 3): which examples to show, which image region to search, and which views to search before tracking.

3.1 Specify the part with examples from other videos

The coupler is a rare concept, and a name such as “trailer coupler” does not reliably identify it. HitchIt therefore specifies the part with boxed examples. For each part, the bank holds examples with varied appearances, viewpoints, and backgrounds, all from videos other than the one being processed.

Retrieve examples by part appearance. A useful example resembles the target part, not merely the scene around it. HitchIt therefore ranks examples by *part matching*. It averages the model’s image features inside each example’s box and compares this average with every local region of the search image’s feature map. An example’s score is the mean of its eight highest cosine similarities, and HitchIt keeps the eight best-scoring examples.

Pair each example with the search image. The model’s exemplar prompt expects the example box and the target in the same image. HitchIt therefore places a retrieved example beside the search image and gives the pair to the model as one input, with a box only on the example (Fig. 3a). It keeps the most confident predicted box on the search-image side. The example’s box thus specifies what

to find, not where to find it. Each of the eight examples forms its own pair, and HitchIt merges the resulting boxes with non-maximum suppression and ranks them by model confidence.

3.2 Search from the trailer to its coupler

In a full-image pair, the search image fills only half of the model’s fixed input, so a small coupler receives few input pixels. Its appearance alone is also not distinctive, but its place in the trailer is: it sits at the tip of the tongue, which extends from the trailer body. The trailer and tongue are larger and easier to find, so they can guide the search to the coupler. HitchIt therefore narrows the search through them (Fig. 3b). It first finds the trailer in the full image, then the tongue inside a crop around each trailer, and then the coupler inside a crop around the tongue. Each stage pairs its crop with examples of its own part; coupler examples, for instance, show a boxed coupler within a crop of its tongue. Each crop gives the next part more of the model’s input and excludes distractors.

HitchIt keeps up to three candidate trailers and rejects tongues and couplers with implausible size or position. For example, a tongue must lie in the lower part of its trailer box, and a coupler may cover at most a quarter of its tongue. When the tongue crop yields no valid coupler, HitchIt searches the full image for a coupler inside the detected tongue, and then searches inside the trailer.

3.3 Identify once, then track

Searching every frame with every example would be costly. With R examples and N frames, it takes NR model calls for a single part, or 4,851 to 17,424 calls on our videos. HitchIt instead searches four evenly spaced frames in the last fifth of the video, where the approaching camera sees the coupler largest (Fig. 3c). Among the valid coupler boxes from these views, it prefers boxes found through a detected tongue over boxes found only in the trailer. It then selects the most confident box, breaking near-ties toward the larger box, and slightly enlarges the box if it is small. The selected box starts the model’s tracker on its frame, and tracking runs to both ends of the video (Fig. 1a). If no view yields a valid coupler, HitchIt outputs no track rather than following a larger part.

4 Experiments

We first compare HitchIt with existing methods. We then isolate the effects of view choice, example retrieval, and part-by-part search, and finally test the complete method on separate videos.

4.1 Experimental setup

Data. We use rear-camera approach videos from the MSU Trailer Database [1]. Our densely annotated set has 20 videos with 49–176 sampled 1920×1200 frames each, 1,784 frames in total. We annotated coupler boxes on 1,136 of these frames and score all methods on them. The remaining frames, all at the start of a video, carry no annotation and are not scored. The coupler covers a median 0.2% of the image, and within a video its size ($\sqrt{\text{box area}}$) varies by a median factor of 6.6 between the 10th and 90th percentiles. Each example bank holds 99 boxed examples from 99 videos outside this set. HitchIt has one bank per part, and its coupler examples are crops around the tongue.

Metrics. Box intersection over union (IoU) is 0 for no overlap and 1 for an exact match. mIoU averages, over annotated frames, the IoU of the best-matching predicted box, counting a missing prediction as zero. AP_{50} and AR_{50} are COCO average precision and recall [24] at IoU 0.5; recall allows one box per frame. We adapt cumulative IoU (cIoU) from referring segmentation [8] to boxes. It divides the total intersection by the total union over all frames, adding the area of unmatched predicted boxes, so oversized and extra boxes lower it. mIoU is our main metric because it credits partial overlap and does not depend on confidence scores, which are identical for all boxes of a track.

Implementation. We build HitchIt on SAM3 [5] and use its exemplar prompts and box-initialized video tracker unchanged. Part matching uses features from SAM3’s frozen Perception Encoder [25]. SAM3 resizes each input to 1008×1008 , so in a full-image pair the search image receives 504 pixels of input width. We convert the tracker’s masks to tight boxes.

Table 1: **HitchIt localizes the coupler more accurately than published baselines.** Results on the 1,136 annotated frames of the 20 densely annotated videos. Tracked rows search the listed views, then track. Text*: the better of one or three target names. Bold: best automatic result. The annotated-box row is a privileged tracking reference, not an automatic competitor.

Method	Prompt	Searched views	mIoU	AP ₅₀	AR ₅₀	cIoU
<i>Text-prompted foundation models</i>						
SAM3 [5], first frame	text	first frame → track	0.038	0.000	0.000	0.106
SAM3 [5], late views	text	4 late views → track	0.059	0.000	0.000	0.099
Grounding DINO-B [3]	text	every frame	0.023	0.000	0.000	0.022
Florence-2-L [6]	text	every frame	0.015	0.000	0.000	0.016
OWLv2-B [4]	text*	every frame	0.067	0.000	0.002	0.097
YOLO-Worldv2-L [7]	text*	every frame	0.020	0.000	0.000	0.013
LISA-7B [8]	text	every frame	0.023	0.000	0.000	0.020
<i>Example-prompted methods: no box in the new video</i>						
PerSAM [9]	boxed examples	4 late views → track	0.036	0.000	0.003	0.014
HitchIt (ours)	boxed examples	4 late views → track	0.627	0.564	0.750	0.632
<i>Oracle reference: annotated box supplied in the test video</i>						
Annotated box	test-video box	1 frame → track	0.683	0.616	0.787	0.759

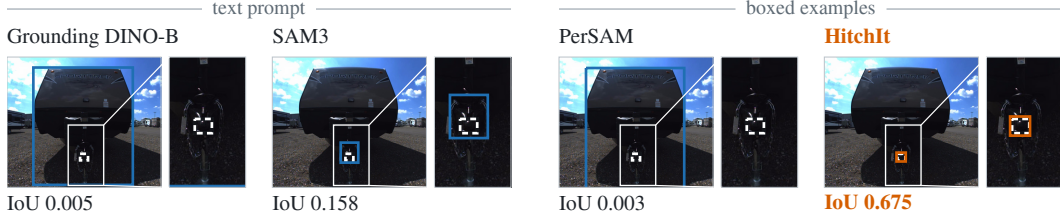


Figure 4: **Finding a larger structure does not identify the coupler.** Grounding DINO-B [3], late-view SAM3 text prompting [5], PerSAM [9], and HitchIt on the same frame, each with a full view and a close-up of the outlined region. PerSAM follows Table 1’s boxed-example and tracking protocol; its box extends below the full view. Blue: baseline predictions; orange: HitchIt; dashed white: the annotated coupler box. Only HitchIt isolates the coupler in this example.

Compared methods. Text-prompted models receive the name “trailer coupler”. Grounding DINO-B [3], Florence-2-L [6], OWLv2-B [4], YOLO-Worldv2-L [7], and LISA-7B [8] process every frame independently and return one box per frame. SAM3 receives the name either as a video prompt on the first frame or as an image prompt on four late views, whose most confident box starts SAM3’s tracker. PerSAM [9] receives eight full-image coupler examples per view, retrieved by part matching and converted to masks, and uses HitchIt’s four late views and tracker. For an oracle reference, as in Fig. 2b, we start the tracker from the annotated box on each video’s last annotated frame.

Ablation baseline. *Direct search* skips the trailer and tongue stages. It retrieves full-image coupler examples by part matching and searches the full image, with HitchIt’s views and tracker.

4.2 Comparison with existing methods

Examples from other videos identify the coupler; names do not. In Table 1, HitchIt reaches 0.627 mIoU. Text-prompted settings reach 0.015 to 0.067, all with an AP₅₀ of 0.000. Even on the same four late views, SAM3 reaches only 0.059. Size alone does not explain the gap: for couplers of at least 80 pixels (517 frames), text prompts reach at most 0.108 mIoU, versus 0.715 for HitchIt.

Baselines can box the wrong structure. Figure 4 shows Grounding DINO-B and PerSAM boxing the trailer. Late-view SAM3 text prompting boxes the hitch assembly, while HitchIt boxes the coupler. PerSAM fails even though it also receives boxed coupler examples from other videos: it reaches only 0.036 mIoU. On 92.5% of frames, Grounding DINO-B returns a box that covers at least half of the coupler but is at least four times its area; its median box is 63 times the coupler’s area. OWLv2 fails

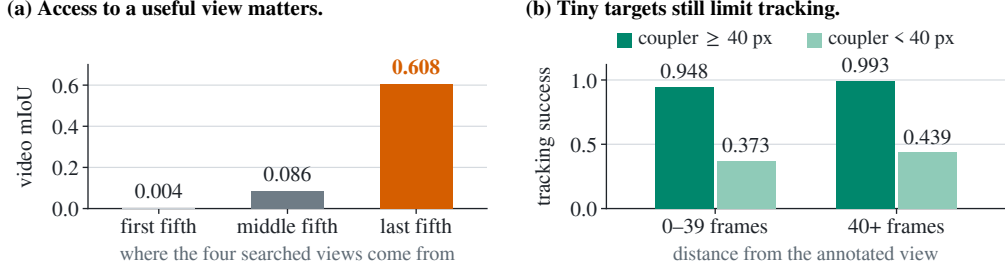


Figure 5: **Late views aid identification; tiny couplers can still limit propagation.** (a) Direct search on four views from the first, middle, or last fifth of each video, followed by the same tracker. (b) With an annotated starting box, the fraction of frames tracked correctly (box IoU ≥ 0.5), grouped by distance from that view in sampled frames and by coupler size.

differently. On 752 of the 1,136 frames, it proposes a box overlapping the coupler with IoU ≥ 0.5 , but its text score selects such a box on only two. A detailed description does not help either. With the prompt “small metal socket at the tip of the trailer tongue”, Grounding DINO-B’s median box shrinks to 1.6 times the coupler’s area, but only two frames reach IoU ≥ 0.5 . Its errors include a frame fitting, a decal, and the vehicle’s hitch ball, so matching the coupler’s scale does not single it out.

4.3 Temporal context: search late views, then track

Late views make the coupler findable. To isolate the choice of views, we fix direct search and the tracker and change only where the four searched views come from: the first, middle, or last fifth of each video. If none of the four views yields a box, no track is produced. In Fig. 5a, mIoU rises from 0.004 for the first fifth and 0.086 for the middle fifth to 0.608 for the last fifth. The coupler is largest near the end of an approach, so late views give the search the most pixels on the part. The final view need not be the best starting point. In the controlled example in Fig. 1a, tracking from the highlighted detection gives 0.780 mIoU, versus 0.456 when starting from the final view.

Identification, not tracking, is the main difficulty. Figure 2b showed that one correct box lets the tracker follow the coupler through a video. In Table 1, the oracle reference, which starts the same tracker from the annotated box, gives 0.683 mIoU across all 20 videos, whereas the name prompt on the late views gives 0.059. Identifying the coupler correctly in a suitable frame is therefore the key step, and HitchIt’s automatic identification reaches 0.627.

Tiny couplers remain hard to track. Figure 5b groups tracking success (IoU ≥ 0.5) from the annotated starting box by distance from that view and by coupler size. For couplers at least 40 pixels in size ($\sqrt{\text{box area}}$), success is 0.948 within 39 sampled frames of the starting view and 0.993 beyond. For smaller couplers, success drops to 0.373 and 0.439, respectively.

4.4 Spatial context: where the gains come from

Table 2 varies how examples are retrieved and where the coupler is searched, with the four late views and the tracker fixed. *Scene matching* ranks examples by the similarity of whole-image embeddings, from CLIP [26] or from SAM3’s encoder, instead of part matching. Two spatial baselines use HitchIt’s cropped coupler examples without trailer or tongue predictions. One searches the full image, and the other searches six fixed, overlapping tiles, each half the image width and two-thirds its height.

Match the part, not the scene. Ranking examples by the part’s appearance gives the best direct-search mIoU (0.608), above whole-scene ranking with CLIP (0.590) or with SAM3 features (0.544).

Part-by-part search beats fixed crops. Part-by-part search reaches 0.627 mIoU. With the same cropped coupler examples, six fixed tiles reach 0.460 while making six times as many model calls as a full-image search. Searching the full image with the cropped examples gives 0.486. Predicted trailers and tongues place each crop around the coupler, which fixed tiles cannot do. Part-by-part search also exceeds direct search with full-image examples in mIoU, AP₅₀, and AR₅₀.

Table 2: **Part matching and part-by-part search improve localization.** All rows use the same four late views of the 20 densely annotated videos and the same tracker. The first group varies example retrieval; the last two compare search regions with cropped coupler examples. Bold: best.

Search setup	mIoU	AP ₅₀	AR ₅₀	cIoU
<i>Direct search: how examples are retrieved</i>				
Scene matching (CLIP [26])	0.590	0.456	0.676	0.587
Scene matching (SAM3 features)	0.544	0.413	0.634	0.542
Part matching (SAM3 features)	0.608	0.512	0.705	0.657
<i>Direct search with cropped coupler examples</i>				
Full image	0.486	0.442	0.607	0.485
Six fixed tiles	0.460	0.360	0.555	0.465
<i>Part-by-part search</i>				
HitchIt: trailer → tongue → coupler	0.627	0.564	0.750	0.632

Table 3: **Part-by-part search improves on direct search across separate videos.** Results on 67 videos, each scored on one labeled late frame using the tracked box. Every test video is excluded from its example bank. Both methods use part matching and the same tracker. This tests cross-video localization, not tracking accuracy across all frames.

Search setup	mIoU	AP ₅₀	AR ₅₀	cIoU
Direct search	0.513	0.349	0.537	0.438
HitchIt (part-by-part search)	0.645	0.718	0.761	0.649

4.5 Evaluation on a wider range of videos

To test HitchIt on a wider range of videos, we use 67 further videos from the MSU Trailer Database. Each has one annotated coupler box on a late frame, 90 to 93% of the way through the video. We split the videos into five groups and process each group with example banks built from the other four groups, which gives 53 or 54 examples per part. HitchIt never receives the labeled frame or its box; we score the box that tracking produces on that frame.

Part-by-part search holds up on separate videos. In Table 3, part-by-part search improves all four metrics over direct search. It raises mIoU from 0.513 to 0.645 and roughly doubles AP₅₀, from 0.349 to 0.718. The mIoU gain is positive in all five video groups, from 0.003 to 0.244.

5 Conclusion

HitchIt localizes a small, rare part whose appearance alone is not distinctive throughout a long, high-resolution video by choosing what a frozen foundation model sees. Boxed examples from other videos specify a part that its name does not identify, and late views show it large. Part-by-part search follows the trailer’s structure to the coupler, giving it more of the model’s fixed input and excluding distractors. A tracker then carries one identification through the recording. Our results show that identification, not tracking, is the main difficulty: from an annotated starting box, the same tracker reaches 0.683 mIoU. On 20 densely annotated videos, HitchIt reaches 0.627 mIoU, whereas text-prompted foundation models and PerSAM reach at most 0.067. On 67 further videos, part-by-part search raises mIoU from 0.513 to 0.645 over full-image search. Tracking very small couplers remains difficult. The same division of work, identifying a target in a few chosen views and regions and then tracking it, may help other long-video tasks that depend on small, uncommon details. We will release our dense coupler annotations.

References

- [1] Yousef Atoum, Joseph Roth, Michael Bliss, Wende Zhang, and Xiaoming Liu. Monocular video-based trailer coupler detection using multiplexer convolutional neural network. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5477–5485, 2017.
- [2] Ashok Dahal, Jakir Hossen, Sumanth Chennupati, Ganesh Sistu, Kazumi Malhan, Muhammad Amasha, and Senthil Yogamani. DeepTrailerAssist: Deep learning based trailer detection, tracking and articulation angle estimation on automotive rear-view camera. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 2339–2346, 2019.
- [3] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *European Conference on Computer Vision (ECCV)*, 2024.
- [4] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. In *Advances in Neural Information Processing Systems*, 2023.
- [5] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliame Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollar, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment anything with concepts. In *The Fourteenth International Conference on Learning Representations*, 2026. URL <https://openreview.net/forum?id=r35c1VtGzw>.
- [6] Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.
- [7] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. YOLO-World: Real-time open-vocabulary object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.
- [8] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. LISA: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.
- [9] Renrui Zhang, Zhengkai Jiang, Ziyu Guo, Shilin Yan, Junting Pan, Hao Dong, Yu Qiao, Peng Gao, and Hongsheng Li. Personalize segment anything model with one shot. In *International Conference on Learning Representations (ICLR)*, 2024.
- [10] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [11] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.
- [12] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2024.
- [13] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019. URL https://openaccess.thecvf.com/content_ICCV_2019/papers/Oh_Video_Object_Segmentation_Using_Space-Time_Memory_Networks_ICCV_2019_paper.pdf.
- [14] Ho Kei Cheng and Alexander G. Schwing. XMem: Long-term video object segmentation with an Atkinson-Shiffrin memory model. In *European Conference on Computer Vision*, 2022.

- [15] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/45c1f6a8cbf2da59ebf2c802b4f742cd-Abstract-Conference.html.
- [16] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018. URL <https://openreview.net/forum?id=H1aIuk-RW>.
- [17] Xudong Wang, Long Lian, and Stella X. Yu. Unsupervised selective labeling for more effective semi-supervised learning. In *European Conference on Computer Vision*, 2022. URL https://www.ecva.net/papers/eccv_2022/papers_ECCV/html/2575-ECCV_2022_paper.php.
- [18] Qing Jiang, Feng Li, Zhaoyang Zeng, Tianhe Ren, Shilong Liu, and Lei Zhang. T-Rex2: Towards generic object detection via text-visual prompt synergy. In *European Conference on Computer Vision (ECCV)*, pages 38–57, 2024. doi: 10.1007/978-3-031-73414-4_3.
- [19] Feng Li, Qing Jiang, Hao Zhang, Tianhe Ren, Shilong Liu, Xueyan Zou, Huaizhe Xu, Hongyang Li, Jianwei Yang, Chunyuan Li, Lei Zhang, and Jianfeng Gao. Visual in-context prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12861–12871, 2024.
- [20] Tianhe Ren, Yihao Chen, Qing Jiang, Zhaoyang Zeng, Yuda Xiong, Wenlong Liu, Zhengyu Ma, Junyi Shen, Yuan Gao, Xiaoke Jiang, Xingyu Chen, Zhuheng Song, Yuhong Zhang, Hongjie Huang, Han Gao, Shilong Liu, Hao Zhang, Feng Li, Kent Yu, and Lei Zhang. DINO-X: A unified vision model for open-world object detection and understanding. arXiv preprint arXiv:2411.14347, 2024. URL <https://arxiv.org/abs/2411.14347>.
- [21] Yang Liu, Muzhi Zhu, Hengtao Li, Hao Chen, Xinlong Wang, and Chunhua Shen. Matcher: Segment anything with one shot using all-purpose feature matching. In *International Conference on Learning Representations (ICLR)*, 2024.
- [22] Miguel Espinosa, Chenhongyi Yang, Linus Ericsson, Steven McDonagh, and Elliot J. Crowley. No time to train! Training-free reference-based instance segmentation. arXiv preprint arXiv:2507.02798v4, 2026. URL <https://arxiv.org/abs/2507.02798v4>.
- [23] Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13084–13094, 2024.
- [24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pages 740–755, 2014. doi: 10.1007/978-3-319-10602-1_48.
- [25] Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, Junke Wang, Marco Monteiro, Hu Xu, Shiyu Dong, Nikhila Ravi, Daniel Li, Piotr Dollár, and Christoph Feichtenhofer. Perception Encoder: The best visual embeddings are not at the output of the network. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2036.
- [26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 8748–8763. PMLR, 2021. URL <https://proceedings.mlr.press/v139/radford21a.html>.