

Back-Tracking from Clarity: Self-Learning to See Text from Afar

Duc Tri Tran¹, Phi Le Nguyen¹*, and Minh Hoai²

¹ Institute for AI Innovation & Societal Impact, Hanoi Univ. of Science & Technology

² Australian Institute for Machine Learning, Adelaide University

Abstract. We propose a self-supervised framework designed to enhance the capability of scene text detectors in identifying and recognizing text in scenarios where instances are shown at significant distances, typically small, blurred, and frequently missed by conventional models. Our approach leverages the high-fidelity performance of existing text spotting models on large, clear text as a foundational supervisor. By temporally back-tracking these high-confidence detections through video sequences, we automatically synthesize pseudo-labels for preceding frames where the distant text is still visually degraded or undersized. These pseudo-labels enable training a student model specialized for early text detection, without requiring any manual annotation. The success of this approach depends on accurate pseudo-label generation, for which we develop a dedicated scene text tracker capable of maintaining consistent text identities across challenging video sequences. In addition, we propose SceneText50, a diverse multilingual outdoor dataset to facilitate training and evaluation. Experiments show that our framework significantly improves early detection accuracy and robustness across varied scenes and languages. Code and data are at <https://github.com/trid2912/BackTrackingText>.

Keywords: Self-learning · Text detection · Text tracking

1 Introduction

The ability to detect and read scene texts from a distance is vital for safety-critical and time-sensitive applications [3, 40]. In autonomous driving [39], for instance, the early recognition of distant traffic signs, lane markings, or warning messages allows vehicles to anticipate hazards well in advance. Similarly, in robotics and intelligent navigation [10, 25], reading directional signs, door labels, or equipment identifiers from afar is essential for efficient path planning and task execution. For assistive technologies, such as wearable devices for the visually impaired, early detection of text at a distance provides users with critical, real-time contextual cues. Despite these needs, existing scene text detectors often struggle when text appears small, blurred, or distant. Their inability to process such challenging instances not only delays perception but also undermines the reliability of downstream reasoning, predictive planning, and overall safety.

* Corresponding author.

The most straightforward strategy to address this is to train a detector using a comprehensive dataset of text across all scales. However, this approach is fundamentally constrained by the high cost and labor-intensive nature of large-scale manual annotation. Furthermore, simply increasing the amount of training data is often insufficient; without explicit focus, models frequently fail to generalize to the specific nuances of small, degraded text, while overly aggressive data augmentation can introduce noise that degrades overall performance. Moreover, the optimal threshold for distinguishable text varies significantly across environments, camera configurations, and linguistic contexts, rendering static, pre-trained models inherently inflexible when deployed in novel domains with distinct visual clutter or signage styles.

To bridge this gap, we propose a novel self-supervised learning framework that enhances small-text detection without the need for manual labels. Our paradigm leverages video sequences where text naturally transitions from small, blurry, and distant instances to larger, clearer ones. By employing a standard detector to identify confident, large-sized text instances and then using a dedicated tracker to propagate these detections backward through time, we automatically generate high-quality pseudo-labels for distant text. These labels allow the detector to iteratively extend its reach and sensitivity to smaller text instances.

Implementing this vision presents two primary hurdles: (1) the lack of robust trackers for distant text and (2) the absence of video datasets capturing these specific distance-varying dynamics. Existing generic object trackers [4, 41] are typically designed for objects with distinct, consistent appearance patterns. They struggle with scene text, which often features repetitive textures and high sensitivity to perspective and illumination. Moreover, these trackers generally operate via a track-by-detection paradigm, which is ill-suited for our backward-propagation requirements where distant instances lack reliable initial detections. Consequently, we require a propagation-based approach that maintains geometric consistency across frames, a task complicated by the fact that small, distant text is often textureless, blurred, or occluded.

To address these challenges, we make three major contributions:

- **A specialized scene text tracker:** We develop a novel tracking mechanism optimized for backward propagation. Rather than tracking the text directly, which is unreliable at distance, we track the parent object “carrier” (e.g., a signpost or vehicle body) which offers more stable textural cues. We integrate this with a generic off-the-shelf tracker such as SAM-2 [26], and introduce a core polygon-based correspondence algorithm. This algorithm approximates each segmented region using a core polygon, allowing us to maintain geometric consistency and infer precise text box coordinates across frames via homography, even when the text itself is degraded.
- **A distance-varying video dataset:** We introduce a new dataset curated for distant scene text detection. Comprising 50 videos and over 440,000 annotated instances across diverse outdoor environments and languages, this dataset provides natural transitions from distant to close-range views. It

serves as a necessary benchmark for evaluating the temporal consistency and early-detection capabilities of future models.

- **A self-supervised learning paradigm:** We introduce a novel framework that utilizes back-tracked pseudo-labels to train more robust detectors. By leveraging naturally occurring scale and clarity variations in videos, our proposed method learns to recognize distant text while maintaining peak performance on clear, large-scale instances, providing a scalable path towards more resilient scene text spotting systems.

2 Related work

Scene text detector. Scene text detection has progressively evolved from detecting horizontal text using rectangular bounding boxes [15, 21, 28] to handling multi-oriented and irregular text through rotated boxes and quadrilateral representations [22, 29, 45, 46]. Despite these advances, most detectors remain biased toward medium and large text instances, as reflected in common benchmarks such as ICDAR and Total-Text [5, 15, 46]. Detecting distant or small text remains challenging due to limited spatial resolution and degraded appearance, a problem related to small-object detection [24]. Existing solutions, including super-resolution approaches [19] or heavy data augmentation, attempt to improve recognition of small objects but often fail to capture the natural scale progression of objects across video frames.

Scene text tracking. Scene text tracking has received comparatively less attention. Recent works employ transformer-based video text spotting frameworks [32] or trackers built on strong image-based spotting models [11, 38]. However, these approaches rely heavily on accurate detections and struggle when text appears very small or visually degraded. Detection-free tracking methods, such as Siamese-based trackers [2, 18], transformer-based trackers [6], or point-based trackers [14], alleviate this dependency but remain sensitive to appearance corruption and template drift. To overcome these limitations, we leverage the capabilities of SAM-2 [26] to track the carrier object of the text and subsequently recover geometric correspondence across frames.

Scene text datasets. While several scene text datasets exist [5, 9, 15, 23, 27, 33, 34, 46], they predominantly focus on clearly visible text and provide limited annotations for extremely small or distant instances. Additionally, these datasets fail to capture the core challenge of early text detection, where text must be annotated from very small and visually degraded states.

Self-improved visual perception. Recent work has explored *self-improvement* strategies that enable perception systems to improve themselves without requiring additional human annotation by exploiting reliable predictions as supervision [17, 35]. For example, scene-adaptive object detection uses high-confidence detections together with tracked object trajectories to generate pseudo-labels for adapting an object detector to a new environment [42, 44]. Similarly, in early event detection, annotations available for complete events have been used to

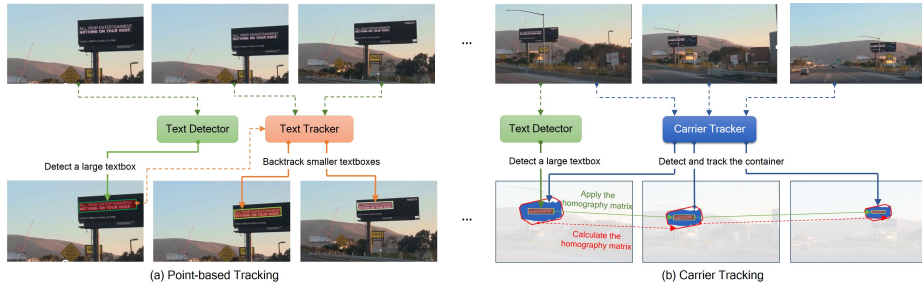


Fig. 1: Our algorithm generates pseudo-labels for small text instances. A large text box is detected and tracked to infer smaller text instances. When texts are too small to track directly, we detect and track their carriers instead. A core polygon is extracted for each carrier, and a homography between consecutive frames is estimated to backtrack small-sized texts.

train models that recognize the same event at an earlier stage [12, 13, 30], while in action understanding, action recognition models have been used to supervise action anticipation networks without requiring separate anticipation annotations [31, 43]. Our work follows the same self-improvement principle but addresses a fundamentally different problem. Rather than adapting to a new scene or anticipating future events, we exploit reliable detections of large, clear text and back-track them to generate supervision for small, distant text, thereby extending the detector’s capability to read text from afar without additional human annotation.

3 Self-Learning Framework

This section presents our self-learning framework for training a scene text detector capable of reading text from afar. The key idea is leveraging a reliable existing detector to automatically generate supervision for smaller, more challenging ones. With sufficiently confident detection of large text in later frames, the model back-tracks these instances through the video to identify their earlier, smaller, and blurrier counterparts. These back-tracked samples serve as pseudo-labels for training a new distant-reading text detector without requiring any manual annotation, as shown in Fig. 1.

3.1 Large-to-small Tracking of Text Instances

The ability to detect large text instances and back-track them to earlier frames where they appear smaller and blurrier is the key component of our framework. For detecting large scene text instances, any reliable text detector can be used. In this work, we employ DeepSolo [38] to produce text masks in our experiments, as its published model has been pretrained on multiple datasets. However, our framework is independent of detector choice and can be applied with any model capable of reliably detecting large, clear texts.

Once confident detections are obtained, the goal is to trace each text instance backward through earlier frames to recover its smaller and blurrier version. This process, referred to as *large-to-small tracking*, forms the foundation for generating high-quality pseudo-labels used in the subsequent unsupervised training stage. The tracking procedure must satisfy two requirements: (1) it must robustly follow text instances across significant changes in scale, perspective, and visibility, and (2) it must maintain approximate geometric correspondence across frames to preserve each instance’s orientation and character order.

To achieve this, we develop a dedicated scene text tracker that leverages a general-purpose tracker to follow the object on which the text is printed and then employs a core polygon-based algorithm to establish correspondences between freeform segments across frames. The boundary of the text instance in the target frame is then recovered by relating the text region and its core polygon in the reference frame to the corresponding polygon in the tracked frame.

Problem Formulation. Let I_t denote an image frame at time step t , where a scene text instance is detected (e.g., by DeepSolo) and represented by a quadrilateral $\mathbf{P}_t = (P_t^1, P_t^2, P_t^3, P_t^4)$, with vertices ordered clockwise. The objective is to back-track this instance to the previous frame I_{t-1} by estimating its corresponding quadrilateral $\mathbf{P}_{t-1} = (P_{t-1}^1, P_{t-1}^2, P_{t-1}^3, P_{t-1}^4)$.

Tracking the carrier object. Unfortunately, directly tracking the text instance itself is highly unreliable, especially when the text becomes small, blurred, or partially occluded in earlier frames. Scene text often exhibits weak texture and repetitive patterns, making feature matching unstable. Our idea to overcome these limitations is to track the carrier object on which the text is printed, such as a signboard or vehicle panel, instead of the text pixels directly. These carrier objects generally provide richer visual cues and more stable boundaries, allowing for robust tracking with generic object trackers, after which the text region can be localized through geometric correspondence within the tracked segment.

Another advantage of the proposed approach is that it allows us to leverage a powerful general-purpose object tracker for our task. In this work, we adopt SAM-2 [26], a state-of-the-art model for prompt-based tracking. We use a single-point prompt placed at the center of the detected text instance $\mathbf{P}_t$ to initiate tracking. Given this prompt, SAM-2 produces segmentation masks M_t and M_{t-1} corresponding to the same prompted point in frame I_t . These masks represent the segmented regions of the carrier object associated with the text instance. However, the segmentation masks are not necessarily quadrilateral or perfectly aligned with the text region we aim to track. Moreover, while M_t and M_{t-1} correspond to the same object, SAM-2 does not provide dense pixel-level correspondences or boundary alignment between the two masks, which are essential for accurately inferring the quadrilateral $\mathbf{P}_{t-1}$ of the tracked text instance.

Geometric correspondence through core polygons. To recover geometric correspondence between text instances across frames, we first fit two n -vertex core polygons, C_t and C_{t-1} , to the segmentation masks M_t and M_{t-1} , respectively (the definition of a core polygon and its fitting procedure will be described in the next paragraph). Each core polygon serves as a convex approximation of



Fig. 2: Geometric correspondence through core polygon at frame t (left) and $t-1$ (right) in case $n=6$. The mapping $\mathbf{H}$ between core polygons C_t and C_{t-1} is used to project the detected text region P_t back to its position in the previous frame P_{t-1} .

the free-form mask boundary, providing a structured and parametric representation of the region. We then use the Random Sample Consensus (RANSAC) algorithm [8] to estimate a homography matrix $\mathbf{H}$ that maps C_t to C_{t-1} . This homography captures the geometric transformation between the two carrier objects and is applied to project the detected text region P_t in frame I_t to its estimated position P_{t-1} in frame I_{t-1} , as shown in Fig. 2. The resulting P_{t-1} preserves the geometric structure and orientation of the text instance while accounting for perspective and motion changes.

Core polygon estimation. One approach to parameterizing a free-form segmentation mask is to utilize its convex hull. However, the number of hull vertices inherently varies across different instances and frames, making them unsuitable for establishing consistent geometric correspondence in video sequences. To address this limitation, we propose an algorithm to approximate the segmentation mask using a polygon with specific structural properties. Intuitively, our approximation satisfies three criteria: (1) the number of vertices is fixed; (2) the orientations of the polygon edges are predefined and constant; and (3) the approximation error between the polygon and the mask is as small as possible. The first two properties are designed to establish a consistent topology between core polygons across successive frames, while the third ensures that the spatial relationship between the core polygon and the enclosed text remains stable over time. Formally, for any n -gon, we define an n -core polygon $\mathcal{P}_n$ as a polygon characterized by the following properties:

Definition 1. For an integer $n \geq 3$, a core polygon $\mathcal{P}$ of a mask is a polygon satisfying:

- It is a circumscribing polygon of the mask (meaning its edges are tangent lines to the mask), and its number of vertices does not exceed n .
- All of its internal angles (excluding those coincident with the mask’s convex hull) are equal to $\frac{(n-2)\pi}{n}$.

In the following, we demonstrate that by increasing the number of vertices n , the approximation error between the core polygon and the mask vanishes, with

their ratio converges to 1. Figure 3 illustrates the effectiveness when increase the number of vertices n .

Theorem 1. *Let $\mathcal{Q}$ be a convex polygon, and let $\mathcal{P}$ be its circumscribing polygon. Denote $P_1, P_2, \dots, P_n$ as the vertices of $\mathcal{P}$ (ordered in the counterclockwise direction). Assume that $P_{r_1}, \dots, P_{r_m}$ are the vertices of $\mathcal{P}$ that are not vertices of $\mathcal{Q}$. We define the Looseness Measure ($\mathcal{L}$) as the ratio of the perimeter of the circumscribing polygon to the perimeter of the convex polygon. Therefore, the Looseness Measure is bounded by*

$$\mathcal{L} = \frac{p_{\mathcal{P}}}{p_{\mathcal{Q}}} \leq \frac{1}{\sin \frac{\rho}{2}}, \quad (1)$$

where $\rho = \min \hat{P}_{r_i}$ is the minimum internal angle of the core polygon.

For a core polygon, all interior angles are equal, so $\rho = \frac{(n-2)\pi}{n}$. As the number of vertices n increases, this constant interior angle approaches π . Consequently, the ratio between the perimeter of the core polygon and that of the mask’s convex hull converges to 1.

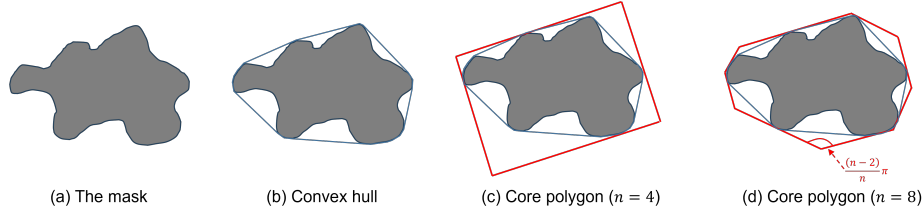


Fig. 3: Illustration of core polygons. A core polygon is the smallest equiangular polygon that circumscribes the mask. By increasing the number of vertices of the core polygon, the distance between the core polygon and the convex hull of the mask can be made arbitrarily small.

3.2 Auto-generated labels and model training

This stage uses the large-to-small tracking framework to generate training labels for distant text and to train a detector capable of reading text from afar.

Label generation. To generate high-quality training labels, we employ a multi-stage pipeline that prioritizes high-confidence detections while ensuring temporal continuity through advanced tracking. The process is initialized using a reliable large-text detector to identify primary text instances. To expand these labels and maintain temporal consistency, we perform back-tracking using two ways:

1. Point-based Tracking: We first utilize CoTracker3 [14] to track the scene text by its bounding points across preceding frames. To ensure label integrity, we only retain tracking results that maintain a confidence score > 0.8 .
2. Carrier Object Tracking: For instances reaching the final frame of the sequence, we further backtrack by identifying the underlying carrier object (the surface or item the text is attached to) using SAM-2 [26].



Fig. 4: Examples of text instances in ArTVideo, DSText, RoadText, and SceneText50 datasets. While ArTVideo focuses on large text and often ignores smaller instances, DSText contains small text that remains largely unchanged in size throughout the sequence. RoadText frequently merges labels of different text together, whereas our SceneText50 dataset uniquely contains small-to-large text tracks.

To resolve potential spatial conflicts, since detections occur independently per frame, we implement a strict priority protocol. Detection masks are prioritized; a back-tracked mask is integrated into the final label set only if it does not significantly overlap with an existing detection. This hybrid approach, combining point-level precision with object-aware segmentation, produces a diverse and dense set of pseudo-labels spanning a wide range of text scales, distances, and motion profiles. Note that this multi-module tracking pipeline and the use of future frames occur *only* during offline pseudo-label generation. Once trained, the student detector runs independently on each frame, without future frames or any tracking module, incurring no additional inference overhead.

Training. We freeze the image encoder, and train the decoder modules using the auto-generated pseudo-labels. To ensure stable convergence and better generalization, we employ Exponential Moving Average (EMA) during training. This adjustment allows the model to capture the fine-grained geometry of small and distant text more effectively than standard sampling densities. All experiments use a single NVIDIA RTX A6000 (48GB), AdamW optimizer, weight decay 10^{-4} , EMA decay 0.999 and batch size 8.

4 SceneText50 dataset

Developing and evaluating methods for distant text detection requires video data in which scene text appears across a wide range of scales—from far, small, and blurry to near and fully legible—within the same trajectory. However, existing video datasets for scene text (*e.g.*, ArTVideo [11], DSText [34], RoadText [27])

largely feature text of relatively consistent size within each sequence, offering limited scale variation and providing few examples of genuinely small or distant text, as shown in Fig. 4. Such datasets are insufficient for studying large-to-small back-tracking, for training models capable of reading text from afar, and for reliably evaluating performance on small text. To address this gap, we collect a new dataset, SceneText50, designed specifically to contain size-varying text instances within their tracks, including many instances of small and distant text. This section describes the data collection process, annotation protocol, and key statistics of SceneText50.

Video data collection. To support distant text detection, our dataset must contain video sequences in which scene text appears across substantial scale changes, from small and barely legible at a distance to large and clear as the camera approaches. Such scale variation naturally occurs in scenarios where a camera is worn or mounted on a moving platform (*e.g.* a person, a robot, or a car). Motivated by these application settings, we specifically search for ego-centric walking and driving videos in which the camera moves through real environments and text gradually becomes larger as the viewer approaches it.

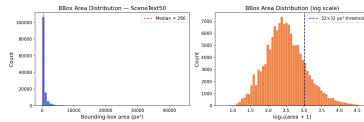
To find such videos, we use YouTube search queries including “4K walking”, “video walking”, “video driving”, and “4K driving”, which return many ego-centric footages with sufficient resolution and forward motion for our task. Although our self-learning framework is not limited to any particular language or script, in this work we focus on scenes containing Latin-alphabet text. This choice is driven by practical considerations: (1) collecting and annotating data across all language families would exceed our available resources, and (2) Latin-script text spans many different languages and is broadly used worldwide, making it a representative and impactful starting point. Importantly, our approach generalizes to non-Latin scripts as well, and future extensions of SceneText50 may include additional languages and regions. Following this data collection procedure, we gather 50 videos, each 1-2 hours long, from 17 different countries across multiple continents (North America, South America, Europe, Asia, Oceania, and Africa).

Data division and annotation. After collecting the videos, we manually divide each long video into short clips of 5-15 seconds. Each clip is selected to contain at least one scene text instance that appears small at first and gradually becomes larger as the camera approaches. For a typical video, this process yields approximately 100-200 clips. From each video, we then uniformly sample about 10% of these clips and aggregate them into our final dataset, resulting in a total of 841 video clips of 5-15 seconds each. We instruct annotators to label every scene text instance appearing in these 841 clips using quadrilateral bounding boxes. Importantly, these annotations are used only for evaluation; our self-learning framework does not require human annotation for training.

To construct the train/test split, we divide the labeled clips of each video based on their temporal order: the first two thirds are designated as training data, and the remaining one third as test data. This split is designed to simulate the real application scenario of on-the-fly self-learning in a new environment. For example, consider a robot navigating a new building or a person wearing

Table 1: Statistics of SceneText50 and comparison with other text video datasets.

Dataset	#Video	#Frame	Annotation	Text size	
				Average	Variance
ArTVideo [11]	60	14.2K	Polygon	3,539	59016
DSText [34]	140	62.1K	Rectangle	1,758	15769
RoadText [27]	1000	300K	Upright rectangle	2,141	24406
SceneText50	841	67.3K	Quadrilateral	985	340411

**Fig. 5:** Distribution of bounding-box areas in SceneText50, skewed toward small text instances.

smart glasses walking through a new city. As they move through the environment, the system has time to observe and adapt through self-learning. After this adaptation period, the system should be able to detect small, distant text more effectively in the same general environment, such as the same city, building, or visual style, which are not necessarily the exact same corridor for a robot or the identical home-to-work route for a person. Our dataset design reflects this intended scenario: the first two-thirds and the last one-third of each video naturally share the same broad environment while still offering different scenes, viewpoints, and lighting conditions.

We use only the ground-truth annotations from the test clips to evaluate the performance of the distant text detector, ensuring that model training remains annotation-free while enabling rigorous and unbiased performance assessment.

Data statistics. SceneText50 contains a total of 67,371 annotated frames, forming 14,263 unique text trajectories. Each trajectory captures a text instance as it transitions from one size and perspective to another. Across all trajectories, the minimum and maximum text areas average 288 and 1,288 pixels respectively, yielding an average scale ratio of 7.94. This large variation, far greater than what is typically seen in existing datasets, indicates that many trajectories undergo substantial scale changes, which is essential for evaluating distant text detection.

Table 1 compares SceneText50 with major video-based scene text datasets. ArTVideo [11] focuses on arbitrarily shaped text but contains relatively few videos. DSText [34] includes dense and small text but covers only six predefined scenarios. RoadText [27] provides large-scale street-view text but generally features text of relatively stable size across trajectories.

In contrast, SceneText50 offers two unique advantages. First, it explicitly targets videos where text appears at small sizes and grows larger as the camera approaches, which is a natural property of outdoor and ego-centric video that existing datasets largely overlook, as evidenced in Fig. 4. Second, it features the smallest average text area among all compared datasets, making it particularly well-suited for studying distant and small-scale text detection. Additionally, SceneText50 also exhibits a high text density per frame, surpassing even RoadText despite covering a broader range of environments. The distribution of bounding-box areas in Fig. 5 further confirms that the dataset is dominated by small text instances.

5 Experiments

5.1 Evaluation settings

Tracking. The tracking evaluation follows a back-tracking paradigm to maintain consistency across the training data. The process is initialized at the last frame of each sequence, where the ground-truth bounding box of the scene text is used as the initial template. The tracker then proceeds to estimate the object’s trajectory in reverse, from the final frame back to the first frame. To ensure a fair comparison between various tracking architectures, all models are initialized using the same standardized bounding box coordinates. This setup minimizes the impact of initialization bias and focuses the evaluation on the tracker’s ability to maintain spatial consistency and handle temporal degradation over the length of the video sequence.

Self-learning to read from afar. We investigate the efficacy of two distinct strategies for small text detection: Synthetic Data Augmentation and Pseudo-Label Generation. To rigorously evaluate these approaches, we establish two experimental tracks: **(1) Detector Consistency:** We train multiple detection architectures under both settings to evaluate their impact on standard detection metrics. Performance is measured by the average Precision and Recall across the testing set to determine which labeling strategy produces more discriminative feature representations. **(2) Early Spotting Performance:** We evaluate the model’s capacity for early detection and early spotting (recognition) within a text-spotting framework. In this setting, the teacher and student models share an identical architecture to isolate the influence of the data-generation strategy. To quantify temporal efficiency, we measure the Average Detection Delay, defined as the number of frames elapsed between the initial appearance of a scene text instance, typically in a blurred, small state, and its first correct identification and recognition at a predefined IoU threshold.

5.2 Experiment metrics

To evaluate tracking performance, we follow LaSOT [7] and employ three primary metrics: Precision@20px, Success@0.5IoU, and Area Under Curve (AUC). Precision@20px measures the percentage of frames where the Euclidean distance between the tracked target’s center and the ground truth is within a 20-pixel threshold. Success@0.5IoU represents the ratio of frames where the Intersection over Union (IoU) between the predicted and ground-truth bounding boxes exceeds 0.5. Finally, the AUC provides a comprehensive assessment of the tracker’s overall robustness by calculating the area under the success curve across all IoU thresholds from 0 to 1.

To evaluate the performance of early detection, we utilize an evaluation framework that measures the trade-off between Average Detection Delay and the False Alarm Rate (FAR), following the protocol from [16]. This approach defines a correct detection based on spatial accuracy-requiring an Intersection

Table 2: Text tracking results on three datasets. The proposed method outperforms several state-of-the-art trackers across three performance metrics. Prec and Succ denote Precision@20px and Success@0.5IoU, respectively, as defined in Sec. 5.2.

Models	SceneText50			ArTVideo			DSText		
	AUC $\uparrow$	S@50 $\uparrow$	P@20 $\uparrow$	AUC $\uparrow$	S@50 $\uparrow$	P@20 $\uparrow$	AUC $\uparrow$	S@50 $\uparrow$	P@20 $\uparrow$
SiamRPN++ [18]	30.24	29.62	57.53	80.65	76.64	63.58	61.82	48.15	42.16
STARK [36]	37.55	39.99	63.80	83.65	73.25	67.53	64.17	52.12	46.40
MixFormer [6]	37.77	38.39	66.48	86.58	80.52	72.02	63.58	52.39	46.76
SeqTrack [4]	37.55	39.14	62.97	88.53	80.11	72.85	65.24	54.02	47.72
ARTrackv2 [1]	36.08	37.28	61.59	85.84	73.50	67.20	66.06	52.78	48.61
SAM-2 [26]	26.43	22.38	56.09	39.48	41.09	39.48	46.27	50.37	44.65
CoTracker3 [14]	38.65	44.01	57.02	90.99	85.75	76.74	80.59	63.59	57.54
Proposed	44.71	49.23	74.62	91.19	86.15	76.83	85.95	68.14	73.62

over Union (IoU) overlap above a specific threshold. FAR represents the proportion of total declarations that are incorrect, serving as a metric for the system’s reliability. Meanwhile, Detection Delay quantifies temporal efficiency by calculating the number of frames elapsed between an object’s first appearance and its initial detection, with a maximum delay penalty applied to any ground truth objects that the system fails to identify.

5.3 Results

Tracking results. Table 2 provides a quantitative comparison of the proposed text tracking method against several state-of-the-art trackers across the SceneText50, ArTVideo, and DSText benchmarks. Notably, on the SceneText50 dataset, the proposed method reaches a Precision of 74.62, representing a significant improvement over the 66.48 achieved by MixFormer. Furthermore, the method demonstrates robust generalization on the challenging DSText dataset, where it secures an AUC of 85.95 and a Precision of 73.62, substantially outperforming the next-best model, CoTracker3, which records 80.59 and 57.54 respectively.

In addition, the qualitative results in Fig. 6 demonstrate that traditional single-object trackers, such as MixFormer, often lose the target as it regresses toward the first frame, where the text is significantly smaller and blurred. Similarly, point-based trackers like CoTracker3 may suffer from point-drift when the local texture of the scene text becomes indistinguishable from the background. In contrast, the carrier-object approach leverages the larger, more stable context of the object carrying the text (the traffic sign in this case). This provides a more reliable spatial anchor, allowing the model to infer the text’s position even when the characters themselves are no longer legible.

Results on self-learning to detect and spot text from afar. Table 3 presents a comparative analysis of scene text detection and spotting performance when models are fine-tuned using two data strategies: our proposed pseudo-labeling framework (via backtracking) and a traditional data augmentation approach. For the latter, we randomly sample five frames from each video in the

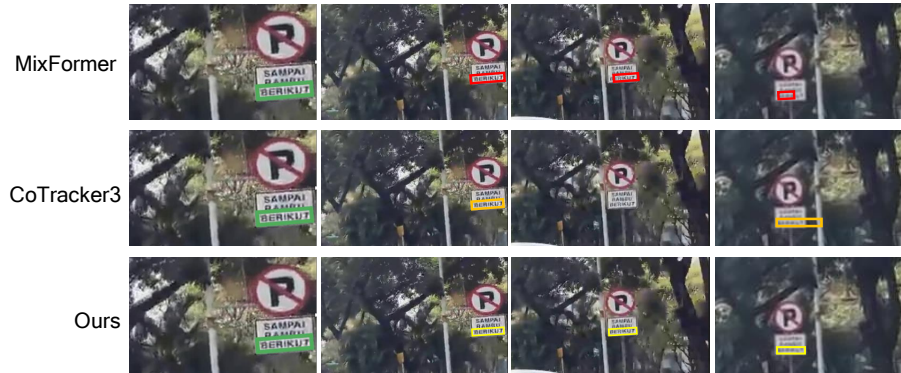


Fig. 6: Qualitative comparison of back-tracking performance. The sequence is initialized using the ground-truth bounding box at the final frame (left, green bbox) and propagated backward to earlier frames. Rows depict the trajectories generated by a single-object tracker (MixFormer), a point-based tracker (CoTracker3), and our proposed approach using carrier-object tracker, respectively.

Table 3: Comparison of different approaches for generating training data to improve the performance of a scene text detector, either using traditional data augmentation or our text-tracking results. We consider three types of detection models: an open-vocabulary object detector (Grounding DINO), a scene text detector (DPText-DETR), and a scene text spotter (DeepSolo). Training with tracked text leads to significantly higher recall while maintaining similar or higher precision.

Models	Training Data by	Precision $\uparrow$	Recall $\uparrow$
Grounding DINO [20]	Augmentation	14.21	29.77
	Text-tracking (proposed)	34.03	34.74
DPText-DETR [37]	Augmentation	44.61	21.80
	Text-tracking (proposed)	44.58	23.12
DeepSolo [38]	Augmentation	52.25	37.75
	Text-tracking (proposed)	51.88	40.27

ArTVideo [11] and DSText [34] datasets, resize them to half of their original size, and apply Gaussian blur to synthesize small-text instances. This process produces a synthetic augmentation dataset consisting of 1,000 images.

Across three distinct architectures, our method consistently yields higher Recall compared with the standard augmentation strategy. This effect is most pronounced for the open-vocabulary Grounding DINO model, where pseudo-labeling more than doubles the Precision from 14.21 to 34.03 and improves Recall from 29.77 to 34.74. These results suggest that the proposed pseudo-labeling method captures a more comprehensive set of text instances, particularly small or blurred ones, thereby improving the model’s sensitivity in scene text detection and spotting tasks.

Furthermore, Figure 7 demonstrates the efficacy of the proposed pseudo-labeling strategy in enhancing the temporal efficiency of scene text systems. In both detection-only and end-to-end spotting scenarios, the pseudo-labeling ap-

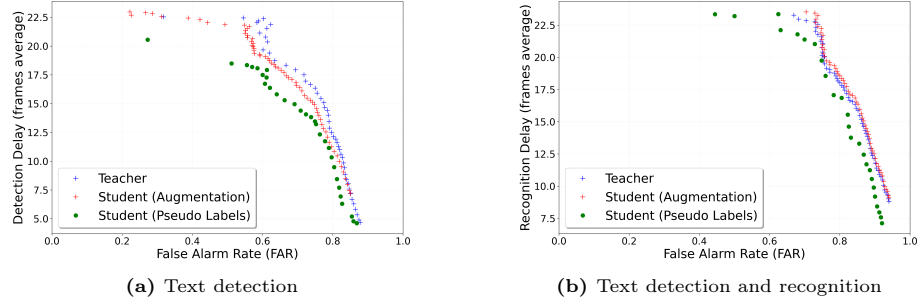


Fig. 7: Evaluation of early detection and early spotting performance. The plots illustrate the trade-off between the False Alarm Rate (FAR) and the average Detection Delay (measured in frames) for both (a) detection-only and (b) end-to-end spotting tasks. We compare the Teacher model (without self-learning) against Student models trained with additional data, either using traditional data augmentation or our proposed pseudo-labeling pipeline. Across both tasks, the student model trained with pseudo-labels (green) demonstrates a stronger ability to identify and recognize scene text at earlier stages of its appearance.

proach shifts the performance frontier toward the lower-left quadrant, indicating that the model can correctly identify and recognize text instances significantly earlier in a video sequence for any given reliability level. This enhanced capability suggests that the backward-propagated pseudo-labels provide more discriminative features for small, blurred, and distant text, allowing the detector to trigger a correct declaration closer to the object’s first appearance frame.

Figure 8 illustrates the superior sensitivity of the pseudo-labeling strategy in long-range scenarios. While the Teacher lacks the discriminative power to detect the distant sign, the Student (Augment) demonstrates improved localization but fails at accurate recognition. In contrast, the Student (Pseudo-labels) model achieves correct end-to-end spotting. This validates that training on backward-propagated labels provides more authentic representations of text degradation, directly supporting the reduced detection delays observed in our quantitative temporal analysis. Note that detection delay measures *observational latency*, i.e., how early the system can first detect and read text, rather than *processing latency* (per-frame inference time). Since our method does not change the detector architecture, processing latency is identical to the base student model, while the observational latency it reduces is what matters for early reading from afar.

Ablation studies. In the setting of carrier-based tracking only, increasing the number of vertices generally improves all metrics: AUC rises from 32% (4 vertices) to 34.6% (16-20 vertices), Success@0.5IoU increases from 33% to 36%, and Precision@20px improves from 56% to 58%. We observe a distinct performance plateau beyond the 16 vertex mark.

The effectiveness of using a carrier object is shown in Tab. 4. Results show that combining point-based and carrier-based tracking components improves significantly across all metrics. This improvement indicates that carrier-based track-

ing effectively serves as a fallback mechanism when point-based tracking fails, helping recover difficult cases and leading to more robust and accurate overall tracking performance.

In addition, we evaluate the impact of the Core Polygon (CP) representation and the RANSAC-based refinement on the tracking performance, as shown in Tab. 5. Using Convex Hull with homography estimation (CH+H) achieves the lowest performance, and RANSAC refinement provides only marginal benefits. In contrast, when the Core Polygon representation is used together with RANSAC refinement (CP+R), the performance improves significantly, demonstrates that the use of CP plays a crucial role in homography matching for carrier tracking.

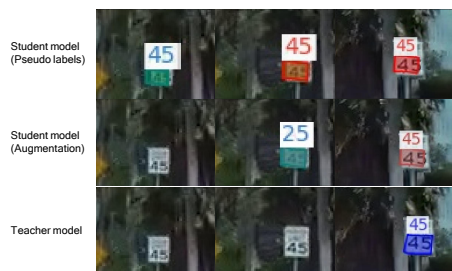


Fig. 8: Qualitative inference comparison of the teacher and student models on distant scene text.

Table 4: Ablation study of tracking with and without carrier.

Tracking approach	AUC $\uparrow$	S@0.5 $\uparrow$	P@20 $\uparrow$
Point-based Tracking	38.62	44.23	57.44
Carrier-based Tracking	34.68	36.00	57.66
Proposed (Full)	44.71	49.23	74.62

Table 5: Impact of Core Polygon (CP) and RANSAC-based refinement, where CH, H, and R denote Convex Hull, Homography, and RANSAC, respectively.

Method	Poly.	RAN.	AUC $\uparrow$	S@0.5 $\uparrow$	P@20 $\uparrow$
CH+H	$\times$	$\times$	37.30	42.07	62.57
CH+R	$\times$	$\checkmark$	37.40	42.10	62.61
CP+R	$\checkmark$	$\checkmark$	44.71	49.23	74.62

6 Conclusion

We have introduced a novel scene text tracker designed to address the problem of large-to-small scene text tracking through a hybrid approach that synergizes point tracking and carrier tracking. Evaluations across multiple scene text datasets demonstrate that our method consistently outperforms current state-of-the-art trackers. Furthermore, we leverage this framework to develop an automated labeling pipeline specifically for distant text instances. Experiments across multiple detectors show that fine-tuning on our auto-generated labels yields superior performance compared to standard augmentation strategies, significantly boosting recall and enabling the model to correctly identify and recognize text instances much earlier in video sequences. Finally, we introduce SceneText50, the first dedicated benchmark for distant-to-close scene text tracking, filling a critical gap in early text detection research.

Acknowledgments. This work is partially supported by VinUniversity under Grant No VUNI.2122.SG04.

References

1. Bai, Y., Zhao, Z., Gong, Y., Wei, X.: Artrackv2: Prompting autoregressive tracker where to look and how to describe. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2024)
2. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional siamese networks for object tracking. In: Proceedings of the European Conference on Computer Vision. pp. 850–865. Springer (2016)
3. Chen, X., Yang, J., Zhang, J., Waibel, A.: Automatic detection and recognition of signs from natural scenes. *IEEE Transactions on image processing* **13**(1), 87–99 (2004)
4. Chen, X., Peng, H., Wang, D., Lu, H., Hu, H.: Seqtrack: Sequence to sequence learning for visual object tracking. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2023)
5. Ch’ng, C.K., Chan, C.S., Liu, C.L.: Total-text: toward orientation robustness in scene text detection. *International Journal of Document Analysis and Recognition* **23**(1), 31–52 (2020)
6. Cui, Y., Jiang, C., Wang, L., Wu, G.: Mixformer: End-to-end tracking with iterative mixed attention. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2022)
7. Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Bai, H., Xu, Y., Liao, C., Ling, H.: Lasot: A high-quality benchmark for large-scale single object tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
8. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Comm. Assoc. Comp. Mach.* **24**(6), 381–395 (1981)
9. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
10. Hassan, B.M., Gamel, S.A., Talaat, F.M.: Attention based unified architecture for arabic text detection on traffic panels to advance autonomous navigation in natural scenes. *Scientific Reports* **15**(1), 34138 (2025)
11. He, H., Ye, M., Zhang, J., Liu, J., Du, B., Tao, D.: Gomatching: A simple baseline for video text spotting via long and short term matching. *Advances in Neural Information Processing Systems* **37**, 25663–25686 (2024)
12. Hoai, M., De la Torre, F.: Max-margin early event detectors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2012). <https://doi.org/10.1109/CVPR.2012.6248012>
13. Hoai, M., De la Torre, F.: Max-margin early event detectors. *International Journal of Computer Vision* **107**(2), 191–202 (2014). <https://doi.org/10.1007/s11263-013-0683-3>
14. Karaev, N., Makarov, Y., Wang, J., Neverova, N., Vedaldi, A., Rupprecht, C.: Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2025)
15. Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., i Bigorda, L.G., Mestre, S.R., Mas, J., Mota, D.F., Almazan, J.A., De Las Heras, L.P.: Icdar 2013 robust reading competition. In: International conference on document analysis and recognition. pp. 1484–1493. IEEE (2013)

16. Lao, D., Sundaramoorthi, G.: Minimum delay object detection from video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
17. Lee, D.H.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: ICML Workshop on Challenges in Representation Learning (2013)
18. Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., Yan, J.: Siamrpn++: Evolution of siamese visual tracking with very deep networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
19. Li, J., Liang, X., Wei, Y., Xu, T., Feng, J., Yan, S.: Perceptual generative adversarial networks for small object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
20. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In: Proceedings of the European Conference on Computer Vision. pp. 38–55. Springer (2024)
21. Lucas, S.M., Panaretos, A., Sosa, L., Tang, A., Wong, S., Young, R., Ashida, K., Nagai, H., Okamoto, M., Yamamoto, H., et al.: Icdar 2003 robust reading competitions: entries, results, and future directions. *International Journal of Document Analysis and Recognition (IJDAR)* **7**(2), 105–122 (2005)
22. Nagy, R., Dicker, A., Meyer-Wegener, K.: Neocr: A configurable dataset for natural image text recognition. In: International Workshop on Camera-Based Document Analysis and Recognition. pp. 150–163. Springer (2011)
23. Nguyen, N., Nguyen, T., Tran, V., Tran, M.T., Ngo, T.D., Nguyen, T.H., Hoai, M.: Dictionary-guided scene text recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2021)
24. Nikouei, M., Baroutian, B., Nabavi, S., Taraghi, F., Aghaei, A., Sajedi, A., Moghaddam, M.E.: Small object detection: A comprehensive survey on challenges, techniques and real-world applications. *Intelligent Systems with Applications* p. 200561 (2025)
25. Raisi, Z., Zelek, J.: Text detection & recognition in the wild for robot localization. arXiv preprint arXiv:2205.08565 (2022)
26. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., et al.: Sam 2: Segment anything in images and videos. In: Proceedings of International Conference on Learning and Representation. vol. 2025, pp. 28085–28128 (2025)
27. Reddy, S., Mathew, M., Gomez, L., Rusinol, M., Karatzas, D., Jawahar, C.: Roadtext-1k: Text detection & recognition dataset for driving videos. In: Proceedings of the IEEE Conference Robotics and Automation. pp. 11074–11080. IEEE (2020)
28. Shahab, A., Shafait, F., Dengel, A.: Icdar 2011 robust reading competition challenge 2: Reading text in scene images. In: International conference on document analysis and recognition. pp. 1491–1496. IEEE (2011)
29. Shi, B., Yao, C., Liao, M., Yang, M., Xu, P., Cui, L., Belongie, S., Lu, S., Bai, X.: Icdar2017 competition on reading chinese text in the wild (rctw-17). In: International conference on document analysis and recognition. vol. 1, pp. 1429–1434. IEEE (2017)
30. Tran, V., Balasubramanian, N., Hoai, M.: Progressive knowledge distillation for early action recognition. In: IEEE International Conference on Image Processing (2021). <https://doi.org/10.1109/ICIP42928.2021.9506507>

31. Tran, V., Wang, Y., Zhang, Z., Hoai, M.: Knowledge distillation for human action anticipation. In: IEEE International Conference on Image Processing (2021). <https://doi.org/10.1109/ICIP42928.2021.9506693>
32. Wu, W., Cai, Y., Shen, C., Zhang, D., Fu, Y., Zhou, H., Luo, P.: End-to-end video text spotting with transformer. *International Journal of Computer Vision* **132**(9), 4019–4035 (2024)
33. Wu, W., Cai, Y., Zhang, D., Wang, S., Li, Z., Li, J., Tang, Y., Zhou, H.: A bilingual, openworld video text dataset and end-to-end video text spotter with transformer. arXiv preprint arXiv:2112.04888 (2021)
34. Wu, W., Zhang, Y., He, Y., Zhang, L., Lou, Z., Zhou, H., Bai, X.: Dstext v2: A comprehensive video text spotting dataset for dense and small text. *Pattern Recognition* **149**, 110177 (2024)
35. Xie, Q., Luong, M., Hovy, E., Le, Q.V.: Self-training with noisy student improves imagenet classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
36. Yan, B., Peng, H., Fu, J., Wang, D., Lu, H.: Learning spatio-temporal transformer for visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2021)
37. Ye, M., Zhang, J., Zhao, S., Liu, J., Du, B., Tao, D.: Dptext-detr: Towards better scene text detection with dynamic points in transformer. In: Proceedings of AAAI Conference on Artificial Intelligence (2023)
38. Ye, M., Zhang, J., Zhao, S., Liu, J., Liu, T., Du, B., Tao, D.: Deepsolo: Let transformer decoder with explicit points solo for text spotting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2023)
39. Yuan, C., Zhang, J., Li, J., Zhuo, L.: Pifttext: Progressive injection fusion for text detection in autonomous driving scenarios. In: IEEE International Conference on Deep Learning and Computer Vision. pp. 1–6. IEEE (2025)
40. Zhang, C., Tao, Y., Du, K., Ding, W., Wang, B., Liu, J., Wang, W.: Character-level street view text spotting based on deep multisegmentation network for smarter autonomous driving. *IEEE Transactions on Artificial Intelligence* **3**(2), 297–308 (2021)
41. Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., Wang, X.: Bytetrack: Multi-object tracking by associating every detection box. In: Proceedings of the European Conference on Computer Vision. pp. 1–21. Springer (2022)
42. Zhang, Z., Hoai, M.: Object detection with self-supervised scene adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2023). <https://doi.org/10.1109/CVPR52729.2023.02068>
43. Zhang, Z., Koraisky, F.M., Hoai, M.: Exemplar-based early event prediction in video. In: Proceedings of British Machine Vision Conference (2021)
44. Zhang, Z., Truong, V.Q., Hoai, M.: Efficiency-preserving scene-adaptive object detection. In: Proceedings of British Machine Vision Conference (2024)
45. Zhang, Z., Zhang, C., Shen, W., Yao, C., Liu, W., Bai, X.: Multi-oriented text detection with fully convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
46. Zhou, X., Zhou, S., Yao, C., Cao, Z., Yin, Q.: Icdar 2015 text reading in the wild competition. arXiv preprint arXiv:1506.03184 (2015)