HIGH PERFORMANCE SPACE DEBRIS TRACKING IN COMPLEX SKYLIGHT BACKGROUNDS WITH A LARGE-SCALE DATASET

Anonymous authors
Paper under double-blind review

ABSTRACT

With the rapid development of space exploration, space debris has attracted more attention due to its potential extreme threat, leading to the need for real-time and accurate debris tracking. However, existing methods are mainly based on traditional signal processing, which cannot effectively process the complex background and dense space debris. In this paper, we propose a deep learning-based Space Debris Tracking Network (SDT-Net) to achieve highly accurate debris tracking. SDT-Net effectively represents the feature of debris, enhancing the efficiency and stability of end-to-end model learning. To train and evaluate this model effectively, we also produce a large-scale dataset Space Debris Tracking Dataset (SDTD) by a novel observation-based data simulation scheme. SDTD contains 18,040 video sequences with a total of 63,247 frames and covers 27,350 synthetic space debris. Extensive experiments validate the effectiveness of our model and the challenging of our dataset. Furthermore, we annotate a real debris observation test set to evaluate the generalization of the model from simulated data to real observation scenarios. SDT-Net achieves a MOTA score of 76% on this set, which demonstrates its transferability to real-world scenarios. Our dataset and code will be released soon.

1 INTRODUCTION

During the past decade, global space activities have exponentially increased, which has raised the potential risk of collisions between debris and spacecraft in near-Earth space Katz (2024); Steindorfer et al. (2025). This has led to significant economic losses, drawing increasing attention to the issue of space debris. The Space Debris Tracking (SDT) is to detect and predict the trajectories of space debris, reducing the collision risks and promoting space industry development. This makes monitoring debris extremely urgent and critical for ensuring the safety of space operations.

SDT is an essential task that includes detecting and predicting the trajectories of space debris. It is a computer vision problem, specifically an object tracking task in astronomical videos, where the goal is to identify and follow space debris across continuous frames for a video. However, traditional methods typically focus on object debris detection rather than tracking. These approaches often rely on techniques such as template-matching methods Liu et al. (2012); Murphy et al. (2017) and morphological operator methods Wei et al. (2018); Jiang et al. (2022). While these methods are capable of detecting debris in individual images, they struggle to associate objects across consecutive frames, making them less suitable for space debris monitoring.

Artificial intelligence (AI) has achieved success in fields such as object detection Zou et al. (2023) and object tracking Javed et al. (2022), due to its ability to process large-scale datasets. Traditional methods have limitations in space debris detection and cannot effectively handle long-term tracking tasks. AI models can make up for this deficiency through efficient data processing and learning methods. However, AI models rely on large-scale, high-quality datasets while astronomy data are extremely limited. So, there is no an effective AI model that could meet the requirements of debris tracking mission for popular telescopes like the Legacy Survey of Space and Time (LSST) Tucker (2023) and Wide-Field Survey Telescope (WFST) Wang et al. (2023).

To address the challenge of limited data in astronomy, we follow the idea of observation-based data simulation in the astronomical field Lin et al. (2021); Peterson et al. (2015); Perrin et al. (2014). The

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

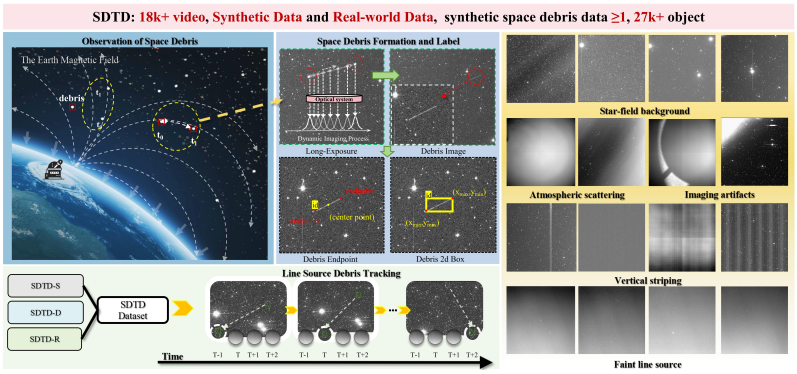

Figure 1: Illustration of space debris imaging and tracking with synthetic endpoint, center, and bounding-box annotations. The right panel shows astronomical backgrounds used in SDTD.

Table 1: Comparison of SDTD with existing optical space-object datasets. Source indicates whether the data are simulated (Sim), collected from real observations (Real), or both (Sim & Real).

Dataset	Source	#Image	#Video	Instances	Object IDs	Tracking	Real Background	Real Test Set
RSONAR Suthakar et al. (2023)	Real	429	3	613	/	✗	✓	✓
SDebrisNet Tao et al. (2023)	Sim	12.8k	47	/	/	✗	✗	✓
ASTA Stoppa et al. (2024)	Real	178	/	/	/	✗	✓	✓
AstroStripeSet Zhu et al. (2024)	Sim	1.5k	/	1.5k	/	✗	✗	✗
StreaksYolo Parisot (2024)	Real	2.4k	/	1.6k	/	✗	✓	✗
BUAA-MSOD Guo et al. (2025)	Real	10k	/	/	/	✗	✓	✓
SpaceDet Xiao et al. (2025)	Sim	20k	4	/	673	✓	✗	✗
SDTD	**Sim & Real**	**63.2k**	**18k**	**95.1k**	**27.3k**	✓	✓	✓

observation-based data simulation is widely proven to lead to realistic data and reliable scientific conclusions, providing a chance for us to design a method for data simulating space debris. Due to its success, we construct a Space Debris Tracking Dataset (SDTD), which is a publicly available dataset designed for space debris tracking. Fig. 1 shows the formation principle, type, examples, and corresponding SDT tasks of space debris. SDTD consists of 18,040 synthetic videos, totaling 63,247 frames and 27,350 annotated debris instances.

In order to apply AI methods to the SDT task, we propose a simple yet effective deep learning method, namely the Space Debris Tracking Network (SDT-Net). We introduce a Debris Aware Enhancemen (DAE) module to highlight debris features, aiding in more accurate detection. A detection module is then used to localize the debris by predicting their positions in the image. After localization, a debris offset module is applied to perform cross-frame data association, generating tracking trajectories for each debris. We further propose a Debris Kinematic Association (DKA) to achieve more efficient and accurate association by exploiting the physical space continuity of space debris motion across frames. In summary, the key contributions of this paper are as follows:

1. We propose a space debris simulation method. Based on this method, we construct SDTD, a benchmark dataset for SDT. It contains 18,040 video data with complex backgrounds and diverse debris targets, and the data is close to the real space debris situation.

2. We introduce the SDT-Net to improve space debris tracking. By integrating a DAE module, a detection module, and a tracking module, we effectively enhance debris representations, leading to more accurate and reliable tracking performance. We further propose a Debris Kinematic Association (DKA) module to strengthen cross-frame association.

3. Evaluation on the constructed SDTD dataset shows that the proposed SDT-Net achieves state-of-the-art performance in terms of space debris tracking on the test set overall.

2 Related Work

2.1 Space Object Datasets and Benchmarks

Existing optical space object datasets have gradually expanded the scope of space-object perception from localization and detection to segmentation and tracking, as summarized in Table 1. Spot-

GEO Chen et al. (2021), RSONAR Suthakar et al. (2023), and SDebrisNet Tao et al. (2023) provide data for satellite or space debris localization and detection, while SPARK Musallam et al. (2021) uses rendered observations to support spacecraft recognition. BUAA-MSOD Guo et al. (2025) further provides real ground-based optical observations with point-level and mask-level annotations for moving space-object detection and segmentation. More recently, SpaceDet Xiao et al. (2025) generates synthetic space-based observations by propagating cataloged resident space objects with an orbital-dynamics model and rendering them through a physical camera model. Its bounding-box and object-identity annotations extend the supported tasks from detection to tracking. However, existing benchmarks remain inadequate for space debris tracking. Real-observation datasets are typically small and limited to single-frame tasks, while the existing tracking benchmark relies on synthetic space-based observations. As a result, current datasets do not jointly provide realistic astronomical backgrounds and diverse long-exposure debris appearances.

3 BENCHMARK DATASET

In this paper, we utilize astronomical images collected by the Zwicky Transient Facility (ZTF) Bellm (2014) to simulate space debris. We propose a simulation tool to generate space debris, which provides sufficient data for constructing the SDTD dataset. Under long-exposure conditions in telescopic observations, debris often appears as a line source in the images. So, our simulation target is to generate realistic long-exposure line sources. We develop a pipeline that includes data preprocessing, line source video generation, and post processing. Fig. 2 and the following paragraphs provide a detailed description of this process.

3.1 DATA COLLECTION AND PRE-PROCESSING

3.1.1 DATA COLLECTION

The Zwicky Transient Facility (ZTF) is a ground-based optical survey project designed to monitor astronomical transient phenomena. It operates on the 48-inch Samuel Oschin Telescope at Palomar Observatory, using a 47-square-degree, 600-megapixel CCD camera for high-cadence, wide-field imaging. Its capability to capture large-scale, high-resolution astronomical images makes it suitable for simulating space debris.

In order to obtain enough data for simulation, we collected 19,235 ZTF astronomical observation data from June 2018 to June 2022. These data contain complex skylight background information, including atmospheric scattered light, moonlight, and human factors such as light pollution. Based on these richer astronomical observation data, we conduct simulations and generated debris videos.

Some regions of the images contain outliers, appearing as zeros or extremely high values, which leads to numerical overflow issues. To ensure the quality of our dataset, we carefully examine the dataset, remove 1,195 unusable images, and retain 18,040 high-quality images.

3.1.2 DATA PRE-PROCESSING

Unlike natural images, astronomical science images typically contain larger pixel values. Therefore, a data processing step is necessary to transform these values into the standard range of grayscale image pixels. To achieve this, we apply the ZScale method Payne et al. (2003) to map image pixel values to a reasonable range. After pre-processing the data, we conduct simulations to generate space debris, providing sufficient data for constructing the SDTD dataset.

3.2 SIMULATION OF SPACE DEBRIS

For a single astronomical image, the Space Debris Simulation generates a video with T frames of line source, simulating the movement of debris across each frame.

3.2.1 LINE SOURCE VIDEO GENERATION

The goal of the space debris simulation is to generate a sequence video with multiple line source. For each debris (set k-th as an example), we represent it as a parameter sequence $\{S_k^1, S_k^2, S_k^3, ...S_k^T\}$ where T is the total frame number. S_k^t is the debris parameters for the t-th frame, which is set as follows:

$$S_k^t = \{\underbrace{x_k^t, y_k^t}_{\text{center}}, \underbrace{l_k, w_k}_{\text{dimension}}, \underbrace{\theta_k, v_k}_{\text{velocity}}\}, \tag{1}$$

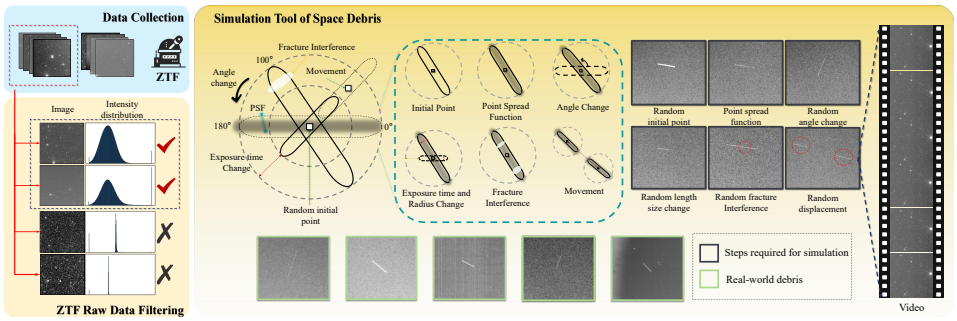

Figure 2: Overview of the space debris simulation pipeline. Step 1: Data Collection. Step 2: ZTF Raw Data Filtering. Step 3: Simulation Tool for Space Debris. Various parameters are applied to simulate space debris. The process generates synthetic video sequences and shows the real-world debris for reference.

where (x_k^t, y_k^t) represents center point coordinates of the line source at time t. l_k, w_k, θ_k and v_k are line length, line width, velocity angle and velocity magnitude, respectively.

With this parametrized line source, we randomly sample a value K as the total line source number and then for the k-th line source, we first generate its S_k^1 by randomly sampling its 6 parameters. The generated S_k^1 provide the initial center point locations, defined dimension, and a fixed velocity of the debris. So, we utilize these parameters to produce subsequent parameters $S_k^2 \sim S_k^T$.

Take the second frame $(t = 2)$ as an example, we calculate the parameters for S_k^2. The center point coordinates (x_k^2, y_k^2) is calculated by combining velocity and initial center as follows:

$$\begin{aligned} x_k^2 &= x_k^1 + v_k \cdot \cos(\theta_k) \cdot \Delta t, \\ y_k^2 &= y_k^1 + v_k \cdot \sin(\theta_k) \cdot \Delta t, \end{aligned} \tag{2}$$

where Δt represents the time interval between consecutive frames, which is set to 1 here for simplification. Thus, the parameters for the second frame are represented as $S_k^2 = \{x_k^2, y_k^2, l_k, w_k, \theta_k, v_k\}$. The same procedure is repeated for each subsequent frame $t = \{3, 4, 5, ..., T\}$. This process continues frame by frame, until all parameters $\{S_k^1, S_k^2, S_k^3, ...S_k^T\}$ are generated.

We proceed to generate the corresponding line source in the image. For each frame t, we take the parameters $S_k^t = \{x_k^t, y_k^t, l_k, w_k, \theta_k, v_k\}$ and follow three steps to generate the line source. With these, at the center point coordinates (x_k^t, y_k^t), we draw a rectangle with dimensions, l_k and w_k, and angle θ_k. Then, we apply random brightness noise to the rectangle. By traversing k, we draw entire K objects and if the rectangle is out of the image, we would neglect it.

At last, using the parameters $\{S_k^1, S_k^2, S_k^3, ...S_k^T\}$, we generate the corresponding images $\{I^1, I^2, I^3, ..., I^T\}$, resulting in a video that captures the movement trajectory of the debris.

3.2.2 Post Processing for Realistic

Since the real line source conforms to the Point Spread Function (PSF) characteristics Kouprianov (2008), we perform point diffusion function processing on the preliminary line source. Additionally, we apply random truncation to the line source to simulate more realistic conditions.

3.3 Dataset Statistics

After simulation, we obtain 18,040 videos with 63,247 frames and 95,081 annotated debris, covering 27,350 unique debris identities. To avoid information leakage, we use 16,040 images to generate the training and validation videos, and the remaining 2,000 images are held out to generate two test subsets. The two test subsets target different debris densities. The SDTD Sparse subset (SDTD-S) contains 1,000 videos with 8,923 frames and 1–2 debris instances per frame, while the SDTD Dense subset (SDTD-D) contains 1,000 videos with 15,810 frames and 1–5 instances per frame.

In addition, we collect and annotate a real-observation subset, termed the SDTD-real subset, to evaluate the generalization ability of debris tracking models. SDTD-Real contains 50 real debris videos with 3,190 annotated frames and covers challenging real-world conditions, including imaging

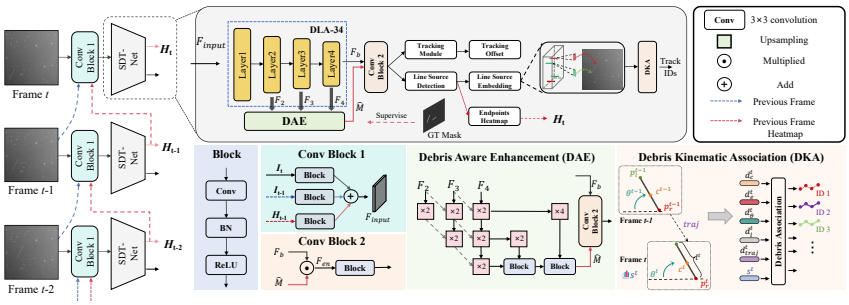

Figure 3: The overall pipeline of our SDT-Net. The inputs are the current frame I_t, the previous frame I_{t-1} and heatmap H_{t-1}. The outputs include amodal endpoints heatmap, line source embedding, and tracking offset.

artifacts such as vertical striping, atmospheric scattering, strong background gradients, faint line sources, and dense star-field backgrounds. More details are provided in the supplementary material.

4 METHOD

4.1 OVERVIEW

The entire pipeline of our Space Debris Tracking Network (SDT-Net) method is shown in Fig. 3, which is a CenterTrack-style Zhou et al. (2020) tracking method. Due to the debris long-exposure as linear, we treat the debris tracking as tracking for linear segments. For each object, we follow the tracking-by-detection idea. Our model detects two endpoints of each line and predicts velocity for cross-frame object association to generate trajectory.

4.2 BACKBONE

For the t-th frame, the model takes three components as inputs, the image of the current frame, the previous frame and its predicted endpoints representing as heatmaps. Each input is passed through a corresponding input-dependent module containing a convolutional layer, a batch normalization layer, and a ReLU activation function. After this, the three feature maps are added together, resulting in a unified feature map $F_{input} \in \mathcal{R}^{W_{in} \times H_{in} \times 3}$. The feature map F_{input} is then passed through our backbone, a DLA model Yu et al. (2018), which consists of an encoder and a decoder. The encoder outputs a series of feature maps at different resolutions, denoted as $\{F_1, F_2, F_3, F_4\}$. The decoder, an IDA up-sampler Yu et al. (2018), takes these feature maps to gradually generate upsampled feature maps and we take the highest resolution one as our backbone feature $F_b \in \mathcal{R}^{W \times H \times C}$.

4.3 DEBRIS AWARE ENHANCEMENT

The Debris Aware Enhancement module first computes a debris mask. Specifically, we introduce another IDA up-sampler, sharing the same structure with the backbone decoder but having independent parameters, as our segmentation head. Its output channels are set to 1 to obtain a debris mask $\hat{M} \in \mathcal{R}^{W \times H \times 1}$ based on encoder features. The output mask is multiplied by F_b as follows:

$$F_{\text{en}} = \hat{M} \odot F_b \tag{3}$$

to acquire the enhanced feature $F_{en} \in \mathcal{R}^{W \times H \times C}$. F_{en} highlights debris features and removes background noise. To ensure that the obtained mask $\hat{M}$ accurately indicates the desired debris region, we use a Segmentation Loss to train the segmentation head. The segmentation mask M are generated by connecting the ground truth endpoints to form a segment and is supervised by the ground truth mask using the following loss function:

$$\mathcal{L}_{\text{seg}} = -\frac{1}{WH} \sum_{i=1,j=1}^{W,H} \Big[M_{ij} \log(\hat{M}_{ij}) + (1 - M_{ij}) \log(1 - \hat{M}_{ij}) \Big], \tag{4}$$

where W, H are image dimensions, M_{ij} is the ground truth, and $\hat{M}_{ij}$ is the predicted probability. The obtained F_{en} is then passed through two modules: a Line Source Detection module and a Debris Offset module.

4.4 LINE SOURCE DETECTION MODULE

In each frame, we detect each line source by detecting its two endpoints. Two endpoints of each line source must be one on the left and one on the right. So, we design our model to produce a left-endpoint heatmap and a right-endpoint heatmap. The left-endpoint heatmap indicates left endpoints of all line sources, where the right one reflects right points. Then, we establish an endpoint embedding module to match different left and right endpoints belonging to a same line source together.

4.4.1 ENDPOINTS HEATMAP

The endpoints heatmap head is a convolutional layer that takes F_{en} as input and feedback heatmaps $\hat{H} \in \mathcal{R}^{W\times H\times 2}$, which has two channels that one for the left and one for the right endpoint. Each pixel in the heatmap represents the likelihood of the position being a line source endpoint.

Follow existing detection heatmap training pipeline, we utilize the Focal Loss Lin et al. (2017) to train the heatmap head as follows:

$$\mathcal{L}_{\text{hm}} = \frac{1}{N_{\text{pos}}} \sum_{whc} \text{Focal}(H, \hat{H}), \tag{5}$$

where $\hat{H}$ and H are the predicted and the ground-truth heatmaps. w, h, c means the spatial and channel index. The ground-truth heatmaps H are generated by drawing Gaussian kernels at the positions of each endpoint.

4.4.2 LINE SOURCE EMBEDDING

The line source embedding predicts features for each endpoint. Specifically, we predict a left endpoint line source embedding maps $\hat{E}_l \in \mathcal{R}^{W\times H\times C_2}$ and a right endpoint line source embedding maps $\hat{E}_r \in \mathcal{R}^{W\times H\times C_2}$. During inference, with the predicted endpoints heatmaps, we first derive local-maximum points for each heatmap, resulting in N left endpoints and M right endpoints. Then we utilize there points to take corresponding embeddings from $\hat{E}_l$ and $\hat{E}_r$, respectively, denoted as $\{e_l^1, e_l^2, ..., e_l^N\}$ and $\{e_r^1, e_r^2, ..., e_r^M\}$. Based on these, we would compute an embedding similarity matrix $S^{emb} \in \mathcal{R}^{N\times M}$ where $S^{emb}_{i,j}$ is computed as follows:

$$S^{emb}_{i,j} = |(e)^i_l - (e)^j_r|. \tag{6}$$

$S^{emb}_{i,j}$ reflects similarity of i-th left point and j-th right point. With this, we provide a simple yet effective scheme, that each left point matches with its nearest neighbors right point to organize an endpoint pair as a detected object.

Inspired by CornerNet Law & Deng (2018), we design the loss function of our embedding as follows:

$$\mathcal{L}_{\text{same}} = \frac{1}{K} \sum_{k=1}^{K} \left[(e_l^k - e_c^k)^2 + (e_r^k - e_c^k)^2 \right], \tag{7}$$

where e_l^k and e_r^k are endpoint features, e_c^k is the mean of e_l^k and e_r^k, utilized to push e_l^k and e_r^k as close as possible. Meanwhile, we introduce $\mathcal{L}_{\text{diff}}$ to separate the features of different pairs as:

$$\mathcal{L}_{\text{diff}} = \frac{1}{K(K-1)} \sum_{k=1}^{K} \sum_{\substack{j=1 \\ j\neq k}}^{K} \max\left(0, \Delta - |e^k - e^j|\right), \tag{8}$$

where Δ is the minimum feature distance threshold and K is the number of annotated objects in the frame. In our experiments, we set Δ to 1. This term facilities to push center features belonging to different objects far away.

4.5 DEBRIS OFFSET MODULE

We construct a Debris Offset module. For the t-th frame, this module predicts two offset maps, $\hat{O}_l^t \in \mathcal{R}^{W\times H\times 2}$ and $\hat{O}_r^t \in \mathcal{R}^{W\times H\times 2}$ for left and right endpoints, respectively.

During inference, the offset plays a key role in data association. Assume that the detection module provides N detected objects, we could index N offsets as: $\{(o_l^t)_1, (o_l^t)_2, (o_l^t)_3..., (o_l^t)_N\}$ and

Table 2: Comparison of trackers on Debris Test and Dense Debris Test. The table compares the performance of various tracking algorithms in two distinct scenarios. n represents the number of debris.

Tracker	SDTD-S						SDTD-D					
	IDF1	MOTA	IDS	HOTA	DetA	AssA	IDF1	MOTA	IDS	HOTA	DetA	AssA
LSD Von Gioi et al. (2012)	11.4	9.8	/	/	/	/	/	/	/	/	/	/
Deepsort Bewley et al. (2016)	71.7	55.3	211	56.4	56.9	55.8	72.9	57.8	1947	58.7	60.1	57.4
Motdt Chen et al. (2018)	69.1	54.0	239	54.2	55.6	52.8	68.6	53.5	2015	53.9	55.7	52.2
CTracker Peng et al. (2020)	83.4	71.0	240	72.9	74.3	71.5	69.3	61.7	1660	58.2	63.9	53.0
Sort+YoloX Ge (2021)	85.6	74.7	**130**	75.1	75.3	74.8	76.0	59.3	1394	61.5	61.7	61.3
CenterTrack Zhou et al. (2020)	86.3	77.1	174	81.0	85.6	75.9	72.5	64.8	1292	66.1	76.7	56.9
ByteTrack Zhang et al. (2022)	91.0	83.3	195	82.8	83.9	81.6	79.4	62.8	1211	66.3	66.8	65.8
OCSORT Cao et al. (2023)	91.6	84.1	138	83.7	84.9	82.5	82.6	68.5	1260	70.8	73.3	68.3
SDTNet (Ours)	**92.1**	**88.1**	157	**88.1**	**91.0**	**85.2**	**85.1**	**76.7**	**896**	**77.7**	**80.8**	**74.1**

$\{(o_r^t)_1, (o_r^t)_2, (o_r^t)_3 ..., (o_r^t)_N\}$. They indicate object endpoint offsets between the current frame and the previous $t-1$ frame. With these predicted offsets, for each detected object, we could derive endpoint locations at the previous frame as follows:

$$\text{left: } (p')_l^{t-1} = p_l^t - o_l^t, \text{ right: } (p')_r^{t-1} = p_r^t - o_r^t, \tag{9}$$

where p' is the derived endpoint coordinate and p is the detected one. These derived endpoints imply the potential previous object locations. With these, we could match objects across frames by measuring the location consistency between the derived endpoints and the detected object endpoints. If the previous frame has M detected objects, we would compute a similarity matrix $S^{obj} \in \mathcal{R}^{N \times M}$ where $S_{i,j}^{obj}$ is computed as follows:

$$S_{i,j}^{obj} = |(p')_l^{t-1} - (p)_l^{t-1}| + |(p')_r^{t-1} - (p)_r^{t-1}|. \tag{10}$$

$S_{i,j}^{obj}$ reflects similarity of i-th object in the t-th frame and j-th object in the $t-1$ frame. Similar to the embedding module, we also utilize nearest neighbor as a matching scheme that the object in the current frame associates with its nearest neighbors in the previous frame. To achieve the aforementioned function, during the training phase, we minimize the difference between the ground-truth offset and the predicted offset as follows:

$$\mathcal{L}_{\text{off}} = \frac{1}{K} \sum_{i=1}^{K} \left| (o^t)_i - [(p^t)_i - (p^{t-1})_i] \right|. \tag{11}$$

4.6 Loss function

SDT-Net is jointly optimized with four objectives corresponding to Debris Aware Enhancement, endpoint detection, endpoint pairing, and cross-frame motion prediction. Specifically, $\mathcal{L}_{\text{seg}}$ supervises the debris mask predicted by DAE, $\mathcal{L}_{\text{hm}}$ supervises the endpoint heatmaps, $\mathcal{L}_{\text{same}}$ and $\mathcal{L}_{\text{diff}}$ learn the endpoint embeddings for line-source assembly, and $\mathcal{L}_{\text{off}}$ supervises the cross-frame endpoint offsets. The overall training objective is

$$\mathcal{L} = \lambda_{\text{seg}}\mathcal{L}_{\text{seg}} + \lambda_{\text{hm}}\mathcal{L}_{\text{hm}} + \lambda_{\text{emb}}\left(\mathcal{L}_{\text{same}} + \mathcal{L}_{\text{diff}}\right) + \lambda_{\text{off}}\mathcal{L}_{\text{off}}. \tag{12}$$

4.7 Debris Kinematic Association

To address missed detections and low-confidence predictions in environments with high debris density and low-SNR line sources, we introduce Debris Kinematic Association (DKA). At test time, we relax the detection confidence threshold to retain candidates that are potentially valid despite their low confidence, which helps mitigate missed detections in complex environments. DKA performs debris id association by integrating the geometric consistency of line sources with kinematic priors, such as predicted orientation and displacement.

We expand the detected endpoints into a debris kinematics representation for association. Specifically, given a detection sequence, for each line source at frame t we denote its two endpoints and confidences as (p_l^t, s_l^t) and (p_r^t, s_r^t), where $s^t > \tau$(τ is the detection confidence threshold). Based on these, we represent each line source by its center c^t, length ℓ^t, direction u^t, and angle θ^t:

$$\mathcal{G}(p_l^t, p_r^t) = \left\{ \tfrac{p_l^t + p_r^t}{2},\ \|p_r^t - p_l^t\|_2,\ \tfrac{p_r^t - p_l^t}{\|p_r^t - p_l^t\|_2},\ \text{atan2}(u_y^t, u_x^t) \right\}. \tag{13}$$

Table 3: Ablation study of SDT-Net and the internal components of DKA.

(a) Ablation study of SDT-Net.

ID	LSDM	DAE	Offset	DKA	SDTD-S / SDTD-D		
					IDF1↑	MOTA↑	HOTA↑
(1)			✓		86.3/72.5	77.1/64.8	81.0/66.1
(2)	✓		✓		90.4/78.5	85.5/64.9	84.8/65.3
(3)	✓	✓			88.2/73.5	86.0/62.9	85.2/63.0
(4)	✓	✓	✓		91.8/80.6	87.7/70.3	87.8/73.6
(5)	✓	✓	✓	✓	**92.1/85.1**	**88.1/76.7**	**88.1/77.7**

(b) Ablation study of DKA components on SDTD-D.

ID	DKA Components						SDTD-D			
	Endpoint	Center	Angle	Length	Score	Traj.	IDF1↑	MOTA↑	HOTA↑	AssA↑
(1)	✓						80.6	70.3	73.6	68.3
(2)	✓	✓					81.0	70.5	73.8	68.5
(3)	✓	✓	✓				82.6	71.7	73.8	70.4
(4)	✓	✓	✓	✓			83.1	72.8	74.5	71.1
(5)	✓	✓	✓	✓	✓		83.4	73.3	74.9	71.5
(6)	✓	✓	✓	✓	✓	✓	**85.1**	**76.7**	**77.7**	**74.1**

We aggregate the endpoint confidence into a single detection score $s = \frac{s_l + s_r}{2}$.

Using the kinematics representation of detections, the association stage solves for optimal matching on the track-detection cost matrix via the Hungarian algorithm Kuhn (1955). For the t-th frame, we define the association cost as:

$$J^t = d_c^t + d_e^t + d_\theta^t + d_\ell^t + (1 - s^t) + (1 - s^t)\, d_{\text{traj}}^t, \tag{14}$$

where $d_c^t = \|(c')^{t-1} - c^t\|/r_c$ is the normalized center distance, where r_c is a hyper-parameter used to normalize the center distance. $d_\ell^t = \frac{|(\ell')^{t-1} - \ell^t|}{\ell^t + \epsilon}$ is the normalized length residual. The endpoint distance d_e is computed in a manner analogous to the center distance by referencing the predicted values and applying normalization.

To suppress spurious candidates introduced by the relaxed threshold, we add a constant-velocity prior based on the previous two frames. At frame t, we derive the velocity of each line source from the center positions observed in the two preceding frames $t-1$ and $t-2$. This velocity facilitates the construction of a motion model that predicts the trajectory of the line source. The predicted center position at time t is denoted as $\hat{c}^t$, and the velocity model as follows:

$$\hat{c}^t = (c')^{t-1} + ((c')^{t-1} - (c')^{t-2}). \tag{15}$$

Then we define the trajectory deviation term:

$$d_{traj}^t = \frac{\|c^t - \hat{c}^t\|}{r_c}. \tag{16}$$

5 Experiments

5.1 Experimental Setup

The entire model is trained for 60 epochs, and the initial learning rate is set to 3e-3, which is decayed by 0.1 at the 20 epoch. We employ the resolution of the input images is set to 1524×1524 and the batch size is set to 2. Besides, the value of λ_{seg} and λ_{emb} are set to 1.0, λ_{hm} is set to 10, and λ_{off} is set to 0.1. We further set r_c to 50. During inference, we first associate detections with $\tau > 0.5$, then match the remaining unmatched tracks using candidates with $0.2 \leq \tau \leq 0.5$.

5.2 Evaluation Metrics

We use standard MOT Bernardin Keni (2008) metrics, including MOTA, IDF1, and IDs, which provide a comprehensive assessment of tracking performance. We also use HOTA Luiten et al. (2021) metrics. Notably, HOTA combines Detection Accuracy (DetA), which measures detection precision, and Association Accuracy (AssA), which evaluates identity consistency across frames.

5.3 Comparison with Existing MOT Methods

As shown in Tab. 2, we compare our proposed SDT-Net with state-of-the-art MOT methods on the SDTD test set. For a fair comparison, all methods were re-trained and evaluated under the same SDTD protocol. Our method outperforms previous approaches in tracking space debris, particularly under dense debris conditions. Notably, SDT-Net achieves a significant performance gain over CenterTrack. Specifically, on the SDTD-S, the model improves MOTA and HOTA by 11.0% and 7.1%, respectively. On the SDTD-D, these improvements increase to 11.9% and 11.6%. Such results demonstrate that SDT-Net provides superior object association and detection robustness, as evidenced by the higher AssA and DetA scores. The consistent superiority of SDT-Net across all primary metrics underscores the effectiveness in complex space debris environments.

Table 4: Comparison with existing trackers and performance of SDT-Net on different subsets of SDTD-Real.

Method	MOTA↑	HOTA↑	DetA↑
CenterTrack	66.9	60.6	70.7
ByteTrack	69.1	64.8	72.1
OCSORT	71.6	69.0	74.5
SDT-Net	**76.0**	**71.4**	**78.4**

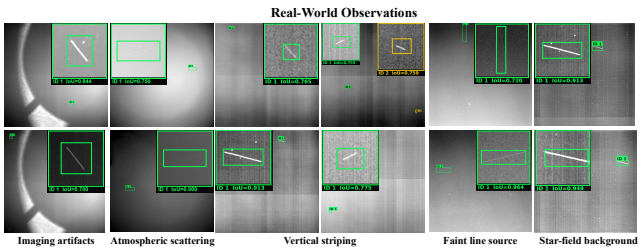

Figure 4: Visualization of sample tracking results from real-world observations.

5.4 Ablation Study

To analyze the contributions of different components in SDT-Net, we conduct an ablation study, as shown in Tab. 3(a).

LSDM: Compared to CenterTrack (row a), introducing line source detection in SDT-Net (row b) brings notable gains across all metrics. For example, MOTA rises from 77.1% to 85.5% on the Debris Test set. This indicates that modeling debris as segments rather than center-based boxes enhances spatial localization, improving detection (higher DetA) and tracking performance.

DAE: Further incorporating DAE (compare rows b and d) enhances the representation of debris features. This results in additional performance gains, with MOTA improving from 85.5% to 87.7% on the Debris Test set and from 64.9% to 70.3% on the Dense Debris Test set. Similarly, HOTA sees a significant increase, confirming that segmentation helps improve tracking robustness.

Offset Module: As shown in rows (c) and (d), introducing the debris offset module alone can significantly improve all indicators. Specifically, MOTA is improved from 86.0% to 87.7% on the debris test set, and from 62.9% to 70.3% on the dense debris test set.

DKA: Adding the DKA module further improves tracking performance, especially on the Dense Debris Test set, where MOTA increases from 70.3% to 76.7% and IDF1 from 80.6% to 85.1%. As shown in Tab. 3(b), progressively incorporating center, angle, length, score, and trajectory cues improves association accuracy and trajectory consistency in dense-debris scenarios.

5.5 Qualitative Results

To further validate the effectiveness of SDT-Net, we evaluate the model on SDTD-Real. As shown in Tab. 4, our method (SDT-Net) achieves a MOTA score of 76%, demonstrating strong performance on real-world data. Fig. 4 presents representative tracking results under several challenging conditions, including imaging artifacts, atmospheric scattering, vertical striping, faint line sources, and dense star-field backgrounds.

6 Conclusion

We introduce SDTD benchmark dataset for tracking space debris. SDTD consists of 18,040 videos with 63,247 frames and 27,350 synthetic debris. Based on this, we propose SDT-Net, which achieves state-of-the-art performance. In addition, we also evaluate the generalization ability of the model through real-world data. The results show that SDT-Net exhibits good tracking performance, proving that the model is transferable to distributed external scenes, ensuring strong tracking capabilities in challenging environments.

AI USE STATEMENT

We have not used generative AI tools for any task that requires disclosure under the ICLR 2027 AI Policy for Authors: no such tool was used to develop the method or its conceptual framework, to formulate hypotheses or claims, to design the experiments, to implement the method, to generate or process data, to translate, or to interpret the results. We used generative AI tools only to edit and rephrase the text of this paper for readability. We have reviewed all AI-assisted edits, and we take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Eric Bellm. The zwicky transient facility. In *The Third Hot-wiring the Transient Universe Workshop*, volume 27, 2014.

Stiefelhagen Rainer Bernardin Keni. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008.

Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, pp. 3464–3468. IEEE, 2016.

Jinkun Cao, Jiangmiao Pang, Xinshuo Weng, Rawal Khirodkar, and Kris Kitani. Observation-centric sort: Rethinking sort for robust multi-object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9686–9696, 2023.

Bo Chen, Daqi Liu, Tat-Jun Chin, Mark Rutten, Dawa Derksen, Marcus Martens, Moritz von Looz, Gurvan Lecuyer, and Dario Izzo. Spot the geo satellites: From dataset to kelvins spotgeo challenge. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2086–2094, 2021.

Long Chen, Haizhou Ai, Zijie Zhuang, and Chong Shang. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In *2018 IEEE international conference on multimedia and expo (ICME)*, pp. 1–6. IEEE, 2018.

Z Ge. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

Yuxi Guo, Junzhe Cao, and Bindang Xue. Multiframe spatio-temporal attention motion-adaptive network for moving space target detection. *Advances in Space Research*, 2025.

Sajid Javed, Martin Danelljan, Fahad Shahbaz Khan, Muhammad Haris Khan, Michael Felsberg, and Jiri Matas. Visual object tracking with discriminative filters and siamese networks: a survey and outlook. *IEEE transactions on pattern analysis and machine intelligence*, 45(5):6552–6574, 2022.

Ping Jiang, Chengzhi Liu, Wenbo Yang, Zhe Kang, and Zhenwei Li. Automatic space debris extraction channel based on large field of view photoelectric detection system. *Publications of the Astronomical Society of the Pacific*, 134(1032):024503, 2022.

JI Katz. Can a satellite dodge space debris? *Acta Astronautica*, 2024.

Vladimir Kouprianov. Distinguishing features of ccd astrometry of faint geo objects. *Advances in Space Research*, 41(7):1029–1038, 2008.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 734–750, 2018.

Bin Lin, Lijun Zhong, Sheng Zhuge, Xia Yang, Yang Yang, Kunpeng Wang, and Xiaohu Zhang. A new pattern for detection of streak-like space target from single optical images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2021.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

Ruiming Liu, Yanhong Lu, Chenglong Gong, and Yang Liu. Infrared point target detection with improved template matching. *Infrared Physics & Technology*, 55(4):380–387, 2012. ISSN 1350-4495. doi: https://doi.org/10.1016/j.infrared.2012.01.006. URL https://www.sciencedirect.com/science/article/pii/S1350449512000084.

Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International journal of computer vision*, 129:548–578, 2021.

Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8844–8854, 2022.

Timothy S Murphy, Marcus J Holzinger, and Brien Flewelling. Space object detection in images using matched filter bank and bayesian update. *Journal of Guidance, Control, and Dynamics*, 40 (3):497–509, 2017.

Mohamed Adel Musallam, Vincent Gaudilliere, Enjie Ghorbel, Kassem Al Ismaeil, Marcos Damian Perez, Michel Poucet, and Djamila Aouada. Spacecraft recognition leveraging knowledge of space environment: Simulator, dataset, competition design and analysis. In *2021 IEEE International Conference on Image Processing Challenges (ICIPC)*, pp. 11–15. IEEE, 2021.

Olivier Parisot. Streaksyolodataset: labeled raw astronomical images for streaks detection, November 2024. URL https://doi.org/10.5281/zenodo.14047944.

Harry Edward Payne, Robert I Jedrzejewski, and Richard N Hook. Astronomical data analysis software and systems xii. *Astronomical Data Analysis Software and Systems XII*, 295, 2003.

Jinlong Peng, Changan Wang, Fangbin Wan, Yang Wu, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pp. 145–161. Springer, 2020.

Marshall D Perrin, Anand Sivaramakrishnan, Charles-Philippe Lajoie, Erin Elliott, Laurent Pueyo, Swara Ravindranath, and Loïc Albert. Updated point spread function simulations for jwst with webbpsf. In *Space telescopes and instrumentation 2014: optical, infrared, and millimeter wave*, volume 9143, pp. 1174–1184. SPIE, 2014.

JR Peterson, JG Jernigan, SM Kahn, AP Rasmussen, E Peng, Z Ahmad, J Bankert, C Chang, C Claver, DK Gilmore, et al. Simulation of astronomical images from optical survey telescopes using a comprehensive photon monte carlo approach. *The Astrophysical Journal Supplement Series*, 218(1):14, 2015.

Michael A Steindorfer, Peiyuan Wang, Franz Koidl, and Georg Kirchner. Space debris and satellite laser ranging combined using a megahertz system. *Nature Communications*, 16(1):1–12, 2025.

F Stoppa, PJ Groot, R Stuik, P Vreeswijk, S Bloemen, DLA Pieterse, and PA Woudt. Automated detection of satellite trails in ground-based observations using u-net and hough transform. *Astronomy & Astrophysics*, 692:A199, 2024.

Vithurshan Suthakar, Aiden Alexander Sanvido, Randa Qashoa, and Regina SK Lee. Comparative analysis of resident space object (rso) detection methods. *Sensors*, 23(24):9668, 2023.

Jiang Tao, Yunfeng Cao, and Meng Ding. Sdebrisnet: A spatial–temporal saliency network for space debris detection. *Applied Sciences*, 13(8):4955, 2023.

Douglas L Tucker. The vera c. rubin observatory legacy survey of space & time (lsst): An astronomical data set for the future. Technical report, Fermi National Accelerator Laboratory (FNAL), Batavia, IL (United States), 2023.

Rafael Grompone Von Gioi, Jérémie Jakubowicz, Jean-Michel Morel, and Gregory Randall. Lsd: A line segment detector. *Image Processing On Line*, 2:35–55, 2012.

Tinggui Wang, Guilin Liu, Zhenyi Cai, Jinjun Geng, Min Fang, Haoning He, Ji-an Jiang, Ning Jiang, Xu Kong, Bin Li, et al. Science with the 2.5-meter wide field survey telescope (wfst). *Science China Physics, Mechanics & Astronomy*, 66(10):109512, 2023.

Yanchao Wang, Dawei Zhang, Run Li, Zhonglong Zheng, and Minglu Li. Pd-sort: Occlusion-robust multi-object tracking using pseudo-depth cues. *IEEE Transactions on Consumer Electronics*, 2025.

Min-Song Wei, Fei Xing, and Zheng You. A real-time detection and positioning method for small and weak targets using a 1d morphology-based approach in 2d images. *Light: Science & Applications*, 7(5):18006–18006, 2018.

Jiaping Xiao, Rangya Zhang, Yuhang Zhang, Lu Bai, Qianlei Jia, and Mir Feroskhan. Spacedet: a large-scale space-based image dataset and rso detection for space situational awareness. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 9402–9410, 2025.

Mingzhan Yang, Guangxin Han, Bin Yan, Wenhua Zhang, Jinqing Qi, Huchuan Lu, and Dong Wang. Hybrid-sort: Weak cues matter for online multi-object tracking. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 6504–6512, 2024.

Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2403–2412, 2018.

Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Motr: End-to-end multiple-object tracking with transformer. In *European conference on computer vision*, pp. 659–675. Springer, 2022.

Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International journal of computer vision*, 129(11):3069–3087, 2021.

Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *European conference on computer vision*, pp. 1–21. Springer, 2022.

Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European conference on computer vision*, pp. 474–490. Springer, 2020.

Zijian Zhu, Ali Zia, Xuesong Li, Bingbing Dan, Yuebo Ma, Enhai Liu, and Rujin Zhao. Sstd: Stripe-like space target detection using single-point supervision. *arXiv preprint arXiv:2407.18097*, 2024.

Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *Proceedings of the IEEE*, 111(3):257–276, 2023.

APPENDIX

A DATASET CONSTRUCTION DETAILS

A.1 DETAILS OF ZSCALE

The ZScale method first calculates the median z_{median} and standard deviation z_{std} of the image pixels. Then uses these two values to estimate a range $[z_1, z_2]$:

$$z_1 = z_{median} - k \times z_{std}, z_2 = z_{median} + k \times z_{std}, \tag{17}$$

where z_1 and z_2 represent the lower and upper bounds of the pixel intensity range, respectively. The hyperparameter k is assigned a value of 2.5. Then, we perform a linear stretch on the grayscale values of the image, mapping the range $[z_1, z_2]$ to $[0, 255]$:

$$I_{scaled}(x, y) = 255 \times \frac{I(x, y) - z_1}{z_2 - z_1}. \tag{18}$$

Finally, pixels with grayscale values below z_1 or above z_2 are set to 0 or 255:

$$I_{\text{final}}(x, y) = \begin{cases} 0, & I_{\text{scale}}(x, y) < 0 \\ I_{\text{scale}}(x, y), & 0 \leq I_{\text{scale}}(x, y) \leq 255 \\ 255, & I_{\text{scale}}(x, y) > 255. \end{cases} \tag{19}$$

A.2 DATASET STATISTICS

As shown in Fig. 5, SDTD contains 18,040 videos with 63,247 frames, 95,081 annotated debris instances, and 27,213 unique object identities. The training-validation set consists of 16,040 videos with lengths ranging from 1 to 4 frames. Its video length distribution is relatively balanced.

The Space Debris Tracking Dataset-Sparse (SDTD-S) contains 1,000 videos with 8,923 frames and 1–2 debris instances per frame. The video lengths range from 1 to 30 frames and are mainly concentrated in relatively short sequences. The distribution reaches its maximum at 4 frames, with 93 videos accounting for 9.3% of the subset.

The Space Debris Tracking Dataset-Dense (SDTD-D) also contains 1,000 videos, including 15,810 frames with 1–5 debris instances per frame. Its video lengths range from 1 to 30 frames and are mainly concentrated between 14 and 18 frames. Compared with SDTD-S, SDTD-D contains more debris instances and generally longer sequences. These characteristics introduce greater ambiguity in cross-frame association and make identity preservation more challenging.

The real-world subset SDTD-R contains 50 videos with 3,190 annotated frames. Its video lengths cover a broader range than those of the simulated subsets. Among them, 18 videos contain 1–20 frames and account for 36.0% of the subset. The remaining videos are distributed across intervals ranging from 21 to 180 frames. This diverse video-length distribution enables the evaluation of SDT-Net on both short and long real-world observation sequences.

A.3 SIMULATION DETAILS AND VISUALIZATION

We provide further details of the debris simulation from two aspects, including the parameter settings used for debris generation and a visual comparison between simulated and real observations.

The simulation parameters and their corresponding ranges are summarized in Tab. 5. These parameters describe the number, initial position, geometric appearance, motion, noise level, and point spread function of the simulated debris, allowing the simulation to generate diverse line-source appearances under different imaging conditions.

To visually compare the simulated debris with real observations, we present representative examples in Fig. 6. The green boxes indicate real debris, while the red boxes indicate simulated debris. As shown in Fig. 6, the simulated debris exhibits visual characteristics similar to those of real line-source debris, including elongated structures, varied orientations, and different lengths and brightness levels.

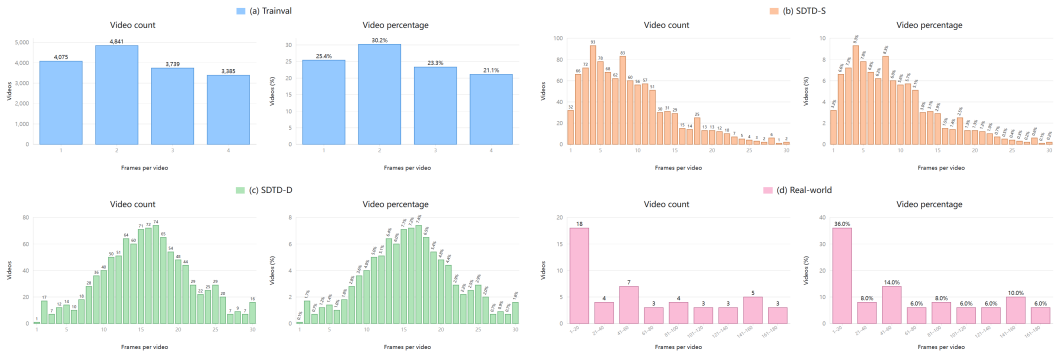

Figure 5: Distributions of video lengths in the training-validation set, the Space Debris Tracking Dataset-Sparse (SDTD-S), the Space Debris Tracking Dataset-Dense (SDTD-D), and the real-world subset SDTD-R. The video count and percentage are reported for each sequence length, while the videos in SDTD-R are grouped into 20-frame intervals.

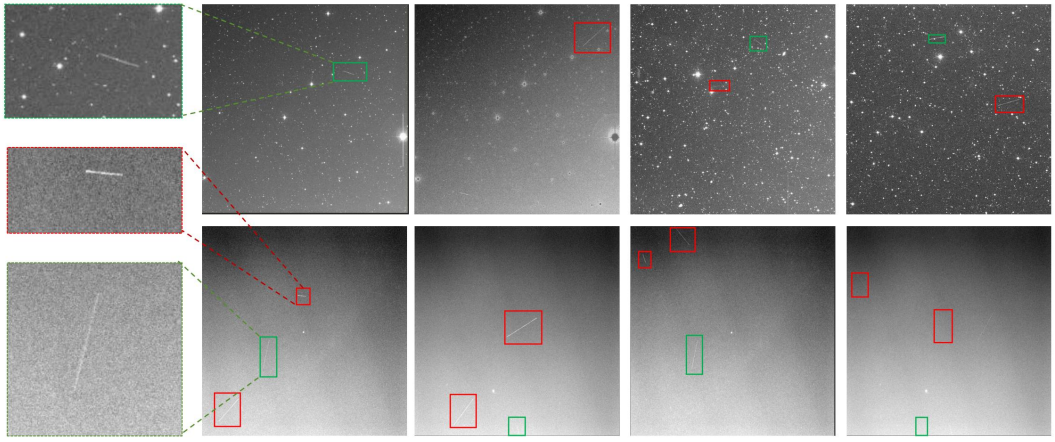

Figure 6: Visual comparison between real and simulated line source debris. Green boxes indicate real debris, while red boxes indicate simulated debris.

Table 5: Table showing variations and their respective parameters.

Variation	**Parameter (Range)**
Number	1, 5
Initial point x	0, 1524
Initial point y	0, 1524
Length	40, 480
Width	12, 32
Angle	0, 360
Fracture	0, 2
Displacement	50, 700
Noise	20, 100
PSF	mean: width/2, δ_{psf}: width/8, S: (20, 120)

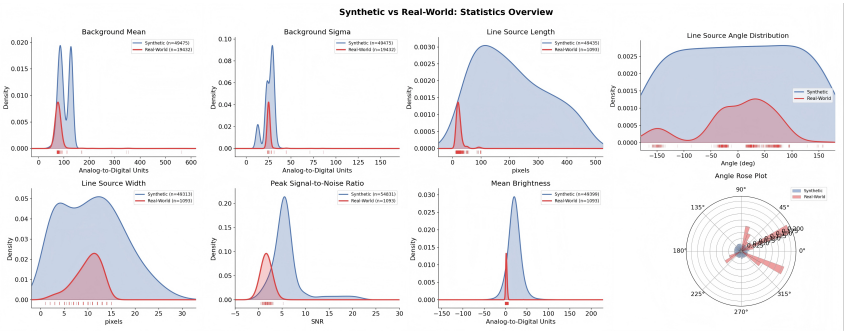

Figure 7: Distribution comparison between SDTD and real observations in terms of background noise, line-source length, width, brightness, signal-to-noise ratio, and orientation.

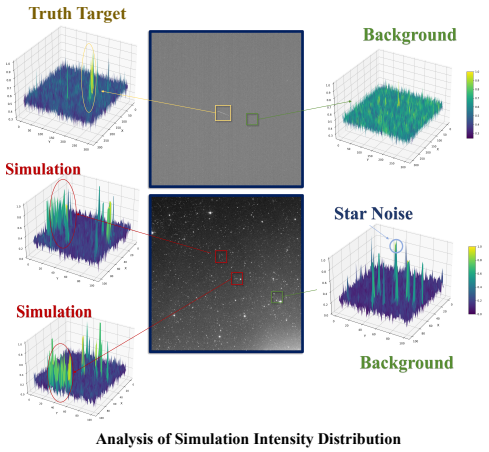

Figure 8: Comparison of the spatial intensity distributions of real debris, simulated debris, and the surrounding astronomical backgrounds.

Table 6: Experimental Environment Setup.

Component	Version
OS	Ubuntu 20.04.5 LTS
Python	3.10.15
PyTorch	2.0.0
CPU	Intel Xeon Platinum 8558
CUDA	11.8

A.4 Distribution Comparison between SDTD and Real Observations

To assess the representativeness of the simulated data, we compare SDTD with real-world observations from both statistical and visual perspectives. The statistical comparison considers background noise, line-source length and width, brightness, signal-to-noise ratio (SNR), and orientation. As shown in Fig. 7, the distributions of real observations largely overlap with, and mostly fall within, those of SDTD, suggesting that the simulation covers the major variations in imaging conditions and debris appearance.

We further visualize the intensity distributions of real and simulated debris embedded in astronomical backgrounds in Fig. 8. The simulated debris exhibits spatial intensity patterns similar to those of real line sources, including the elongated brightness distribution and gradual intensity decay around the central trajectory. These comparisons show that the observation-based simulation reproduces the main geometric and photometric characteristics of real debris observations.

A.5 DETAILS OF DKA

Let i index a track and j a current detection. We back-project each endpoint as $(p'_{q,j})^{t-1} = p^t_{q,j} - o^t_{q,j}$, where $q \in \{l, r\}$. Primed centers, lengths, and orientations are derived from these projected endpoints. The geometric costs compare the detection and track at frame $t-1$:

$$\begin{aligned} d^t_{c,ij} &= \frac{\|(c'_j)^{t-1} - c_i^{t-1}\|_2}{r_c}, \\ d^t_{e,ij} &= \frac{1}{2r_c} \sum_{q\in\{l,r\}} \|(p'_{q,j})^{t-1} - p_{q,i}^{t-1}\|_2, \\ d^t_{\theta,ij} &= \frac{1 - \cos(2\Delta\theta^t_{ij})}{2}, \\ d^t_{\ell,ij} &= \frac{|(\ell'_j)^{t-1} - \ell_i^{t-1}|}{\ell_i^{t-1} + \epsilon}, \end{aligned} \tag{20}$$

where $\Delta\theta^t_{ij} = (\theta'_j)^{t-1} - \theta_i^{t-1}$, $r_c = 50$, and $\epsilon > 0$ stabilizes the denominator. The angular cost accounts for the 180° ambiguity of line orientation.

For tracks observed in both preceding frames, the motion prediction and total cost are

$$\begin{aligned} \hat{c}_i^t &= 2c_i^{t-1} - c_i^{t-2}, \\ d^t_{\mathrm{traj},ij} &= \frac{\|c_j^t - \hat{c}_i^t\|_2}{r_c}, \\ J^t_{ij} &= d^t_{c,ij} + d^t_{e,ij} \\ &\quad + d^t_{\theta,ij} + d^t_{\ell,ij} \\ &\quad + (1 - s_j^t) + (1 - s_j^t) d^t_{\mathrm{traj},ij}, \end{aligned} \tag{21}$$

where $s^t_j = (s^t_{l,j} + s^t_{r,j})/2$. Historical track centers are unprimed and distinct from back-projected detection centers.

We first associate tracks with detections satisfying $s_j^t > 0.5$, then associate the remaining unmatched tracks with candidates satisfying $0.2 \le s_j^t \le 0.5$. Detections below 0.2 are discarded. Hungarian assignment enforces one-to-one matching. The score-weighted trajectory term penalizes motion deviations more strongly for low-confidence candidates. For a fixed set of assigned detections, the additive term $(1 - s_j^t)$ does not change their assignment among tracks.

B EXPERIMENTS AND VISUALIZATIONS

B.1 EXPERIMENTAL SETTING/DETAILS

We ensure reproducibility by providing the experimental environment and computational resources. Tab. 6 shows the environment configuration.

All experiments are performed using the mmdetection framework with Pytorch. The SDT-Net model is trained on 8 NVIDIA 4090 GPUs. The entire model is trained for 60 epochs, and the initial learning rate is set to 3e-3, which is decayed by 0.1 at the 20 epoch. Specifically, we first crop image patches of size 1524×1524 from the original high-resolution images as ground truth. We employ the resolution of the input images is set to 1524×1524 and the batch size is set to 2. Besides, the value of λ_{seg} and λ_{emb} are set to 1.0, λ_{hm} is set to 10, and λ_{off} is set to 0.1. Notably, the tracking methods compared in **Tab. 2** of **Sec. Experiments** in the main manuscript are all retrained using SDTD according to the official settings.

B.1.1 REAL-TIME ANALYSIS

To evaluate the efficiency of SDT-Net, we measured the inference time for a single image. The results are shown in Tab. 7. The model achieves an inference time of 0.34 seconds per image.

Table 7: Inference time of each method on a single image.

Method	Time (s)
CTracker	1.41
Sort+YoloX	0.15
CenterNet	0.37
ByteTrack	0.81
OCSORT	0.28
SDT-Net	0.34

Table 8: False positives (FP) and false negatives (FN) on the debris and dense debris test sets.

Test Set	**FP**	**FN**
SDTD-Sparse	333	596
SDTD-Dense	1698	1928

Table 9: Comparison with transformer-based trackers, joint detectionembedding models, and recent tracking methods on the SDTD-Sparse and SDTD-Dense test sets.

	SDTD-Sparse				**SDTD-Dense**			
Method	MOTA	HOTA	DetA	AssA	MOTA	HOTA	DetA	AssA
MOTR Zeng et al. (2022)	70.4	68.8	72.1	65.5	56.8	54.9	56.5	53.3
TrackFormer Meinhardt et al. (2022)	77.0	71.2	76.5	68.6	60.2	61.3	67.5	54.6
FairMOT Zhang et al. (2021)	72.5	70.2	74.8	66.1	52.4	55.0	59.2	53.5
PD-SORT Wang et al. (2025)	78.9	80.1	82.4	78.0	64.8	67.3	69.1	64.2
Hybrid-SORT Yang et al. (2024)	84.5	85.6	85.1	86.0	68.7	72.1	74.5	69.7
SDT-Net	88.1	88.1	91.0	85.2	76.7	77.7	80.8	74.1

B.1.2 Statistics of False Positives and False Negatives

To provide a clearer understanding of SDT-Net's error behavior, we additionally report the number of false positives and false negatives on both test sets. These statistics complement the tracking metrics presented in the main paper and offer a more detailed information of detection errors in different debris densities.

As shown in Table 8, the dense debris test set exhibits a higher number of false positives and false negatives. These statistics complement the tracking metrics presented in the main paper and provide more detailed information about detection errors under different debris densities. SDTD-D produces more false positives and false negatives in absolute numbers, partly because it contains more frames and debris instances than SDTD-S.

B.1.3 Additional Comparisons with Tracking Models

To provide a more comprehensive comparison across modern tracking paradigms, we additionally evaluate transformer-based trackers, joint detection embedding models, and several recent tracking methods introduced after 2024. The results are summarized in Table 9.

B.1.4 Loss and Backbone ablation experiment

To further verify the sensitivity of the loss weights, we conducted additional experiments by changing the hyperparameters (λ_{hm}, λ_{seg}, λ_{emb}, and λ_{off}). As shown in the tables below, our default setting achieves the best performance:

Tab. 11 compares different backbone architectures. DLA34 achieves the best results, outperforming ResNet50 and Hourglass-104 in both detection and association metrics, demonstrating its superior suitability for debris tracking.

Table 10: Sensitivity analysis of the loss-weight hyperparameters in SDT-Net on SDTD-Sparse. Default values are underlined.

Hyperparameter	Value	MOTA ↑	HOTA ↑	DetA ↑	AssA ↑
λ_{hm}	0.1	77.7	77.8	80.1	75.5
	1.0	85.4	84.3	85.2	83.4
	<u>10.0</u>	88.1	88.1	91.0	85.2
λ_{seg}	0.1	84.4	83.1	86.9	79.3
	0.5	86.8	85.7	88.2	83.2
	<u>1.0</u>	88.1	88.1	91.0	85.2
λ_{emb}	0.1	85.1	85.5	88.7	82.3
	<u>1.0</u>	88.1	88.1	91.0	85.2
λ_{off}	1.0	86.9	87.3	90.1	84.4
	<u>0.1</u>	88.1	88.1	91.0	85.2

Table 11: Comparison of different backbone architectures for SDT-Net on SDTD-Sparse. DLA34 achieves the best overall performance.

Backbone	MOTA ↑	HOTA ↑	DetA ↑	AssA ↑
ResNet50	82.6	80.1	86.6	74.6
Hourglass-104	86.4	85.6	88.7	83.5
DLA34 (ours)	88.1	88.1	91.0	85.2

Table 12: Cross-domain evaluation of SDT-Net on the Space Debris Tracking Dataset-Dense (SDTD-D) and its variants generated using backgrounds from HST, SDSS, and ZTF.

Data Source	MOTA ↑	HOTA ↑	DetA ↑	AssA ↑
HST (F814W)	66.7	64.5	67.2	61.9
SDSS (r-band)	67.4	69.4	71.9	66.9
ZTF (g-band)	69.7	72.1	74.9	69.5
SDTD-D (dataset)	76.7	77.7	80.8	74.1

B.1.5 Cross-Domain Evaluation

To further evaluate transferability under different observational setups, we synthesized three additional datasets using backgrounds from HST, SDSS, and ZTF. We directly applied the pre-trained SDT-Net for inference. The results are summarized in Tab. 12. These results show that SDT-Net retains reasonable tracking performance when evaluated on previously unseen astronomical background sources.

B.1.6 Analysis on Clip Length and Long-Sequence Training

To examine whether training on very short video clips restricts the model's ability to learn temporal cues, we conduct an additional experiment using longer training sequences. Most training and validation clips contain only 1 to 4 frames, while the test sets include sequences that can reach 30 frames, and the dense test set typically contains sequences with an average length close to 18 frames.

We additionally generated a set of long-sequence debris videos for training, consisting of 2,500 videos with 30,500 frames (average length 12 frames). To keep the overall training scale comparable to the original setting, we mixed these with 3,500 videos randomly sampled from the original training set. The combined dataset contains 6,000 videos with 38,760 frames, which is similar in size to the original training set (37,829 frames). We re-trained SDT-Net using this combined dataset and summarize the results in Table 13.

Table 13: Comparison of SDT-Net trained with the original short-sequence training set and the long-sequence-augmented training set on the Space Debris Tracking Dataset-Dense (SDTD-D).

Training Setting	**MOTA** ↑	**HOTA** ↑	**DetA** ↑	**AssA** ↑
Original training set	76.7	77.7	80.8	74.1
Long-sequence augmented set	78.8	79.4	83.4	75.7

We observed a consistent improvement in performance for dense and long segments. In our future work, we will expand the dataset in this way to further improve model performance.

B.1.7 Results visualization

We provide a detailed visualization of the results. As shown in Fig. 9, we first compare the tracking results of SDT-Net with those of other representative methods on SDTD. Compared with CenterTrack and OC-SORT, SDT-Net detects the debris more accurately and maintains more consistent object identities across consecutive frames, demonstrating its effectiveness in debris detection and association.

As shown in Fig. 10 and Fig. 11, we present the tracking results of SDT-Net on real-world data. Both Exp1 and Exp2 demonstrate strong tracking performance on debris, indicating that our model exhibits good generalization capabilities in real-world scenarios.

Furthermore, as shown in Fig. 12 and Fig. 13, we visualize the results on the SDTD test set, providing further evidence of the model's robustness under different test conditions.

B.1.8 Discussion of Integration with Astronomical Processing Pipelines

We discuss how the outputs of SDT-Net can be directly integrated into standard astronomical data-processing pipelines for debris-related tasks. It is straightforward to integrate our system into existing astronomical processing pipelines. In the current debris processing pipeline, a debris-detection module should take a FITS (an image format in astronomy) image as input and output the sky coordinates of each detected object. Our model follows the same interface: it is trained on raw FITS-domain images, so it has the ability to directly operate FITS images. And its predicted pixel-level locations can be converted into standard sky coordinates (Right Ascension and Declination) using the WCS (World Coordinate System) metadata stored in every FITS file. This means the outputs of SDT-Net are immediately usable by downstream astronomy modulessuch as orbit estimation, trajectory prediction, and catalogue updatingwithout any additional transformation or engineering effort.

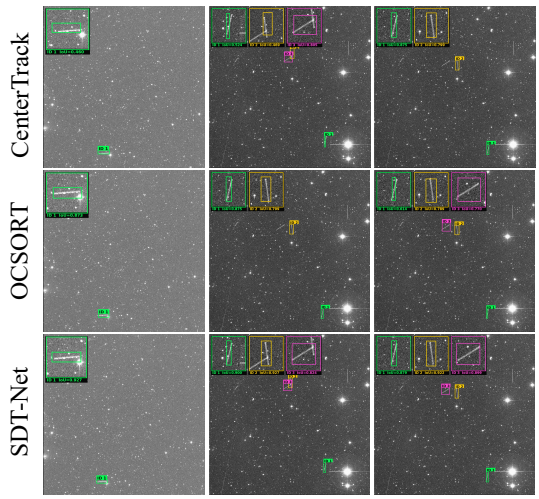

Figure 9: Visualization-based comparison of SDT-Net with other methods.

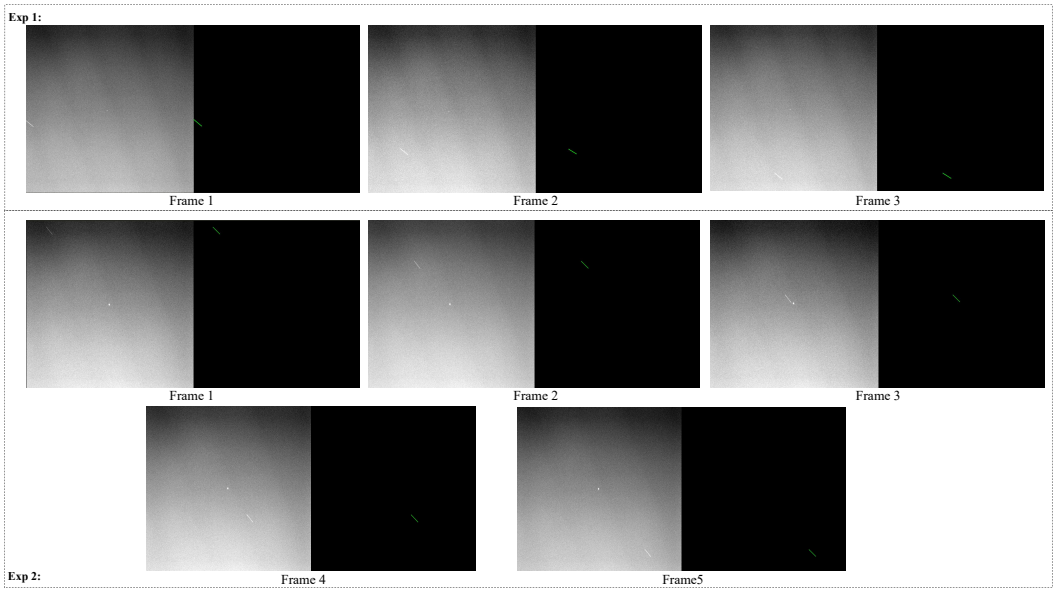

Figure 10: SDT-Net tracking results in the real-world data.

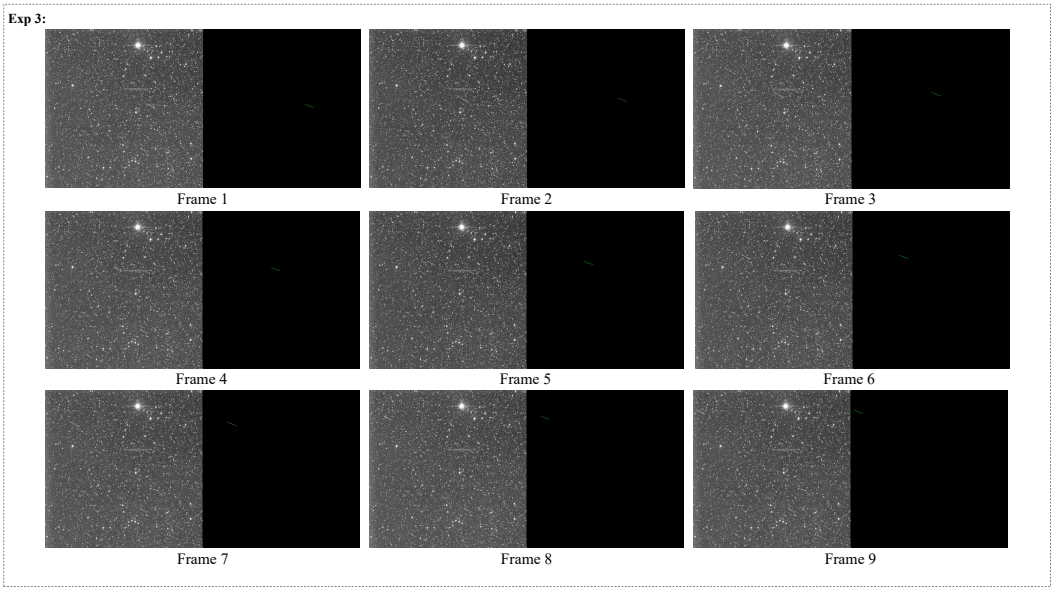

Figure 11: SDT-Net test tracking result example in SDTD.

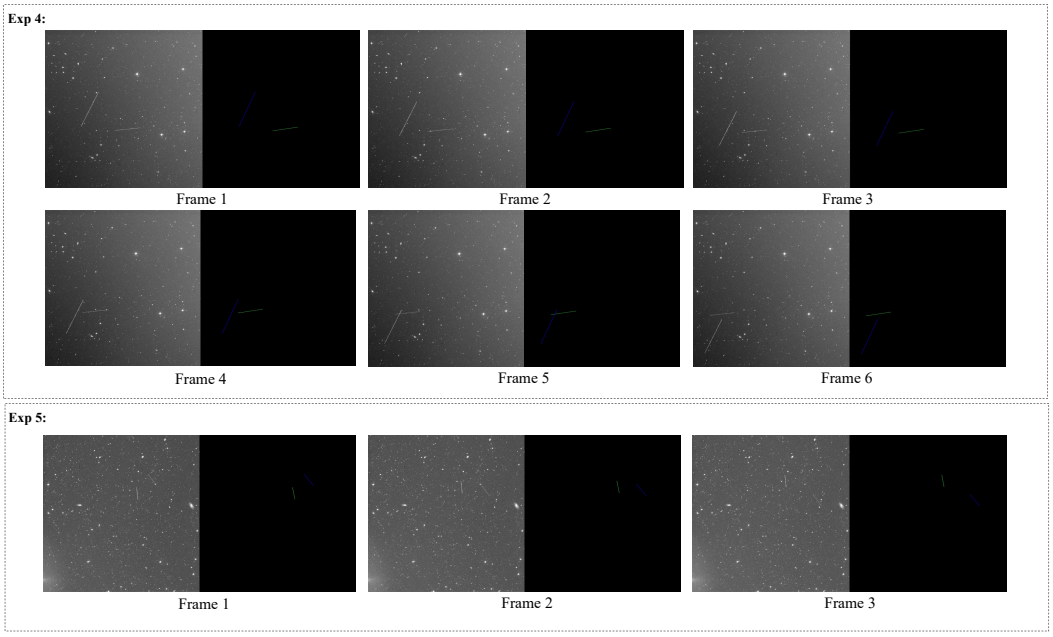

Figure 12: Example of tracking results of SDT-Net on the test set.

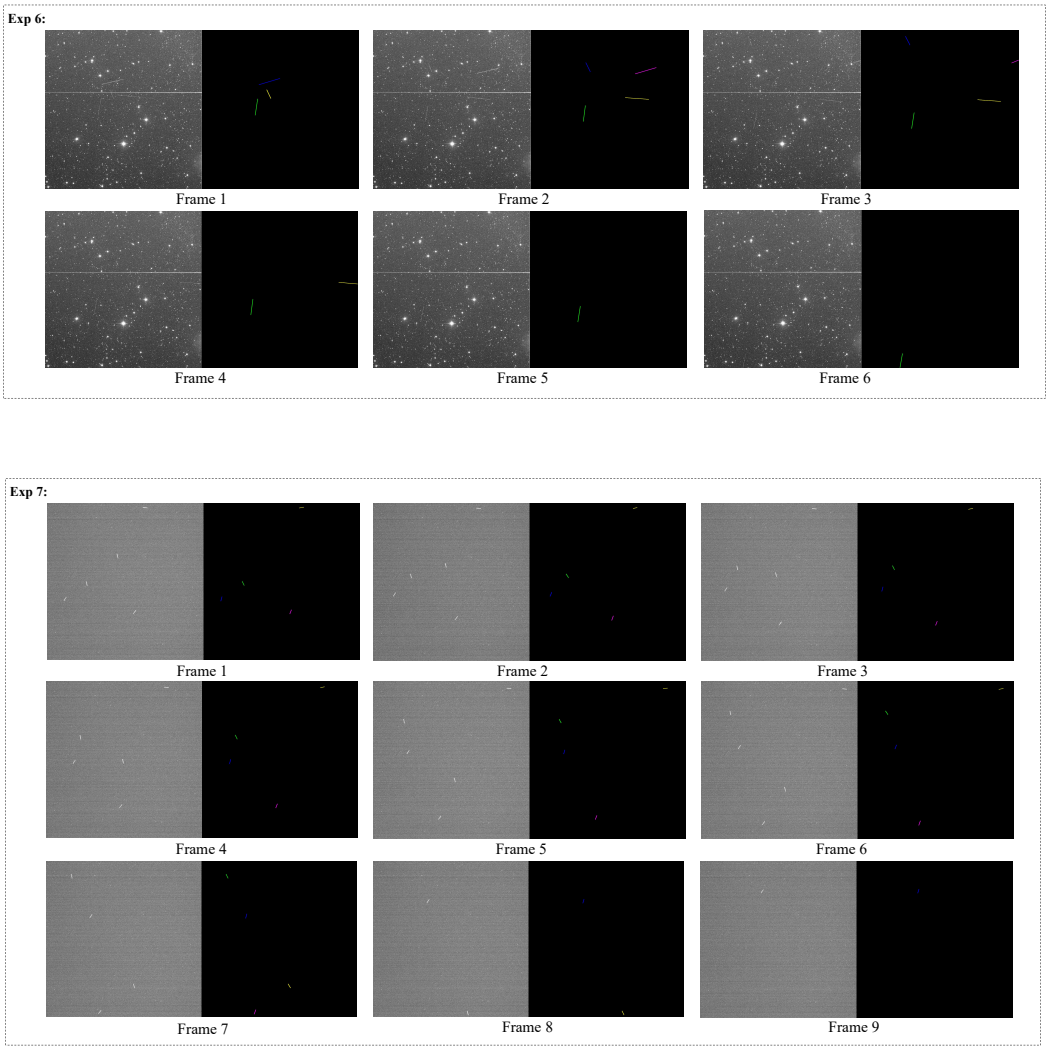

Figure 13: Example of tracking results of SDT-Net on the dense test set.