# GOT-EDIT: GEOMETRY-AWARE GENERIC OBJECT TRACKING VIA ONLINE MODEL EDITING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Human perception for effective object tracking in a 2D video stream arises from the implicit use of prior 3D knowledge combined with semantic reasoning. In contrast, most generic object tracking (GOT) methods primarily rely on 2D features of the target and its surroundings while neglecting the 3D geometric cues, making them susceptible to occlusion, distractors, and large variations in geometry and appearance. To address this, we present GOT-Edit, a novel approach that integrates visual geometry knowledge into a generic object tracker through online model editing. Our approach first leverages the features from the Visual Geometry Grounded Transformer (VGGT), which is trained on large-scale 3D-annotated data and can provide reliable geometric cues directly from a set of 2D images. To tackle the challenge of seamlessly combining geometry and semantics, we develop an online model editing strategy with a null-space constraint that adaptively fuses geometric information while preserving the tracker's learned semantic knowledge, yielding consistently better performance across diverse scenarios than a naive fusion of both cues. Extensive experiments on multiple GOT benchmarks demonstrate that GOT-Edit achieves superior robustness and accuracy, particularly under occlusion and clutter, establishing a new paradigm for combining 2D semantics with 3D geometric reasoning for generic object tracking.

## 1 INTRODUCTION

Generic object tracking (GOT) (Bhat et al., 2019; Li et al., 2019; Javed et al., 2022) aims to track an arbitrary user-specified target object, identified by its initially bounding box in the first frame, and to predict the locations of this target in subsequent frames. However, learning a robust tracker from limited visual information remains a significant challenge, especially in adverse conditions like partial occlusion, cluttered scenes with distractors, and significant object deformations.

Most contemporary GOT trackers are trained on 2D datasets, e.g., (Muller et al., 2018; Fan et al., 2019; Huang et al., 2019; Peng et al., 2024). As a result, their 2D-based representations limited their ability to reason about contextual relationships between a target and its surroundings, such as distinguishing a target under partial occlusion or separating it from background distractors. In contrast, incorporating 3D information provides geometric cues for object boundaries, enabling more precise reasoning to mitigate challenges such as partial occlusion and inter-object discrimination.

Although several studies (Tan et al., 2025a;b; Chen et al., 2025b; Feng et al., 2025; Hu et al., 2025; Xu et al., 2025b; Zhang et al., 2024a) have attempted to leverage 3D information for enhanced tracking, they often rely on additional 3D data, such as objects represented in RGB-D or backgrounds in point clouds. This reliance is impractical, as GOT is primarily performed on 2D video streams. Humans, by contrast, can track targets from the background, near or far, even when observing only 2D videos or single images. This is because our prior 3D knowledge allows for perception that extends beyond the flat image plane (Koch et al., 2018; Gregory, 1997).

Emerging techniques in geometric 3D vision (Wang et al., 2024; 2025a; Zhang et al., 2025; Wang et al., 2025b; Yang et al., 2025) offer a promising direction for advancing GOT. Among these, we adopt the Visual Geometry Grounded Transformer (VGGT) (Wang et al., 2025a) for its strong performance and generalization, in alignment with the GOT objectives. Given one or a few 2D images as input, VGGT learns features for camera pose, point map, and depth estimation, as well as point tracking, While VGGT has shown effectiveness in point tracking (Karaev et al., 2024),

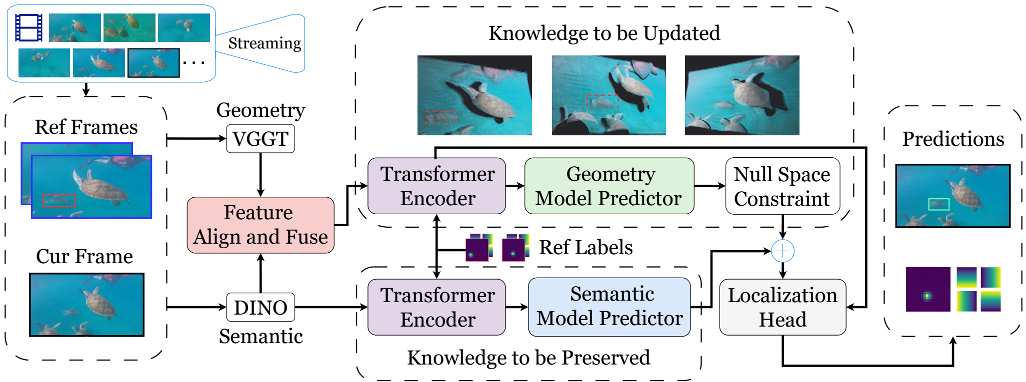


Figure 1: **GOT-Edit** online adapts a tracker using 2D semantic knowledge complemented with 3D geometry knowledge through tracking model editing with a null-space constraint.

perception from 2D semantics remains essential for GOT. This is because point tracking operates at the pixel level and does not require an understanding of object semantics, whereas a robust GOT tracker benefits from both geometric and semantic information.

While geometric information is potentially beneficial for GOT, effectively balancing its contribution with crucial or even dominant semantic information remains a key challenge. As evidenced by our later experiment, a naive fusion strategy improves geometry attributes in tracking but degrades semantic attributes. To address this issue, we propose a novel online model editing technique that better integrates 3D geometric features from VGGT with 2D semantic features (Oquab et al., 2023) for GOT. Our approach is inspired by the *null-space model editing* from AlphaEdit (Fang et al., 2025), which is designed to introduce new knowledge into the null space of a trained model while preserving previously learned knowledge for optimal performance. However, AlphaEdit performs offline model editing, whereas GOT requires online updates to handle dynamically varying targets and backgrounds in both seen and unseen scenarios. To bridge this gap, we develop an online editing technique that enables a tracker to adaptively complement 2D semantics with 3D geometric features.

As illustrated in Figure 1, our system begins by extracting both semantic and geometric features from the current and reference frames. These features are then aligned and fused to create an enriched representation, which serves as new knowledge for online tracker adaptation. Built upon the ToMP (Mayer et al., 2022), our approach employs two model predictors: one for the semantic branch and one for the geometric branch. These predictors generate the model weights for the localization head. During tracking, the reference labels, which provide correspondences for the reference frames and act as few-shot examples of previously predicted and observed information, are dynamically updated to guide a tracker toward the target object. This process guides the model predictors to forecast model weights for the current frame in an online manner. Namely, the semantic model predictor estimates the semantic weights, while the geometry predictor generates complementary weights. A null-space constraint is applied before combining these two sets of weights to preserve the semantic information. Finally, the combined model weights are used by the localization head to localize the target in the current frame.

Our main contributions are threefold. First, we integrate semantic and geometric knowledge into generic object tracking without relying on additional 3D input data. This integration enriches 2D tracking with geometry-aware reasoning, strengthening target discrimination in complex environments. Second, we propose an online model editing method with a null-space constraint, which adaptively incorporates additional 3D geometric knowledge into GOT without degrading the dominant semantic features. Finally, extensive experiments on multiple benchmarks validate the effectiveness of our approach, demonstrating that it unlocks most of the geometric knowledge lacking in existing 2D trackers, resulting in superior performance.

## 2 Related Work

**Generic Object Tracking.** Existing methods for this task are typically derived from two pipelines (Javed et al., 2022): matching-based trackers and tracking-by-detection trackers. The matching-based paradigm formulates tracking as a similarity learning task followed by matching (Bertinetto et al., 2016; Li et al., 2018; Guo et al., 2020; Xu et al., 2020; Voigtlaender et al., 2020; Yu et al., 2020; Zhang et al., 2020; Yan et al., 2021a; Chen et al., 2021; Ye et al., 2022; Guo et al., 2022; Cai et al., 2023; Gao et al., 2022; He et al., 2023; Zhou et al., 2023; Li et al., 2023; Chen et al., 2023; Jinxia et al., 2024; Bai et al., 2024; Shi et al., 2024; Cai et al., 2024; Song et al., 2023; Wu et al., 2023; Zhao et al., 2023; Zhang et al., 2024b; Xie et al., 2024; Zhu et al., 2025). These methods focus on training a deep network to learn a function that can distinguish and match a template of the object to a search region in the current frame. This trained network is then used for tracking. Recent matching-based trackers (Guo et al., 2025; Xu et al., 2025a; Li et al., 2025; Kang et al., 2025; Xie et al., 2025) further improve their robustness by propagating chronological contextual information from predicted hidden states.

Another paradigm, tracking-by-detection, frames generic object tracking as an online detection task (Bolme et al., 2010; Henriques et al., 2012; Kiani Galoogahi et al., 2017; Yao et al., 2018; Lukezic et al., 2017; Danelljan et al., 2017; 2016; Nai & Chen, 2023). Recent trackers under this paradigm employ a model predictor that generates a target-specific tracking model from paired reference images and labels, allowing more accurate object localization in the current frame (Bhat et al., 2019; Mayer et al., 2022; Chen et al., 2025a). The model predictor is dynamically updated for each incoming frame by referring to previous tracking results and hence enhances the tracker's robustness and adaptivity. A separate localization head then uses this updated model to pinpoint the target.

Despite progress in the above two paradigms, they remain limited by their reliance solely on two-dimensional spatial and structural knowledge. To overcome this, our method integrates 2D semantic information with 3D geometric features, enabling a 2D tracker to exploit 3D geometry information through online tracking model editing.

**3D Features for Tracking.** Existing trackers that utilize 3D features fall into two primary categories: those that augment RGB images with additional modalities (RGB+X) (Yan et al., 2021b; Yang et al., 2022; Zhu et al., 2023; Hou et al., 2024; Cao et al., 2024; Tan et al., 2025a;b; Chen et al., 2025b; Feng et al., 2025; Hu et al., 2025) and those that operate directly on point cloud data (Wu et al., 2024; Nie et al., 2024; Liu et al., 2024; Zhang et al., 2024a; Xu et al., 2025b). These approaches require auxiliary inputs during tracking, such as pre-computed depth maps or scene point clouds, which are generally unavailable in real-world scenarios where scenes and objects may be arbitrary and even previously unseen. Another line of research (Doersch et al., 2022; Harley et al., 2022; Doersch et al., 2023; Wang et al., 2023; Karaev et al., 2024), known as point tracking, explores tracking any pixel. Recent extensions (Xiao et al., 2025; Rajič et al., 2025; Wang et al., 2025a; Harley et al., 2025) incorporate 3D information for point tracking, but their formulation remains pixel-level and cannot leverage object-level semantics, which are crucial for GOT.

In contrast, our tracker adaptively integrates 3D geometric knowledge with 2D semantic knowledge for GOT through online model editing. Specifically, we embed VGGT (Wang et al., 2025a) into the 2D tracker, where a sequence of RGB frames is used to derive complementary 3D information. While geometric features from VGGT have proved effective for point tracking (Wang et al., 2025a; Karaev et al., 2024), our method departs from this approach by embedding these features into a 2D GOT tracker via model editing, thereby establishing a direct connection between 3D geometric representations and object-level semantics for tracking.

# 3 Method

Geometric information inferred from 2D image streams benefits GOT by enabling a tracker to move beyond flat representations, but it must be balanced with semantic knowledge. Driven by this insight, we aim to enhance object tracking with geometry-aware reasoning while preserving semantic discrimination. We first introduce null-space model editing in AlphaEdit and explain how it links geometry and semantics. We then justify the track-by-detection paradigm as a natural fit for model editing. Finally, we provide a step-by-step description of our pipeline, highlighting our online model editing approach and objective function.

## 3.1 Preliminary

### 3.1.1 Null-Space Constrained Knowledge Editing

Model editing updates the knowledge stored in a model by adjusting its learned weights. Among existing model editing algorithms, we adopt the AlphaEdit (Fang et al., 2025) because it excels at fusing unbalanced features while avoiding catastrophic forgetting. AlphaEdit treats the feed-forward network (FFN) as a linear associative memory, where input features serve as keys and are mapped to output features through model parameters $\mathbf{W} \in \mathbb{R}^{d_b \times d_a}$:

$$\mathbf{V} = \mathbf{W}\mathbf{K}, \text{ where } \mathbf{K} = [\mathbf{k}_1 \mid \mathbf{k}_2 \mid \ldots \mid \mathbf{k}_u] \in \mathbb{R}^{d_a \times u} \text{ and } \mathbf{V} = [\mathbf{v}_1 \mid \mathbf{v}_2 \mid \ldots \mid \mathbf{v}_u] \in \mathbb{R}^{d_b \times u}. \quad (1)$$

In Eq. 1, $u$ is the number of features to be updated, $d_a$ and $d_b$ are the dimensions of the respective FFN layers, and $\mathbf{k}_i \in \mathbb{R}^{d_a}$ and $\mathbf{v}_i \in \mathbb{R}^{d_b}$ jointly represent the $i$-th key-value pair.

One representative optimization objective for model editing is defined by:

$$\boldsymbol{\Delta} = \arg\min_{\tilde{\boldsymbol{\Delta}}} \left( \left\| (\mathbf{W} + \tilde{\boldsymbol{\Delta}})\mathbf{K}_1 - \mathbf{V}_1 \right\|^2 + \left\| (\mathbf{W} + \tilde{\boldsymbol{\Delta}})\mathbf{K}_0 - \mathbf{V}_0 \right\|^2 \right), \quad (2)$$

where $\mathbf{K}_0$ and $\mathbf{V}_0$ represent originally learned knowledge, while $\mathbf{K}_1$ and $\mathbf{V}_1$ encode newly introduced knowledge. This objective seeks an optimal perturbation $\boldsymbol{\Delta}$, obtained by optimizing over candidate perturbations $\tilde{\boldsymbol{\Delta}}$, to edit the model to account for both original and new knowledge.

In practice, new edits often degrade performance on the learned knowledge, as original associations are disrupted. AlphaEdit addresses this by introducing a null-space constraint: the perturbation $\boldsymbol{\Delta}$ is required to lie in the *null space* of $\mathbf{K}_0$, i.e., $\boldsymbol{\Delta}\mathbf{K}_0 = \mathbf{0}$. It follows that

$$(\mathbf{W} + \boldsymbol{\Delta})\mathbf{K}_0 = \mathbf{W}\mathbf{K}_0 = \mathbf{V}_0. \quad (3)$$

This additional constraint ensures preservation of the learned knowledge when adapting the model to new knowledge. Thus, AlphaEdit is highly suitable for our proposed GOT-Edit, where dominant 2D semantic features serve as the learned knowledge to be preserved, while auxiliary 3D geometric features represent the newly introduced knowledge. Specifically, the tracker predicts the semantic model weights online and the perturbation weights from 3D features concurrently. These geometry-aware perturbation weights are projected into the null space of the semantic knowledge for semantics preservation. The semantic weights and the projected perturbation weights are then combined, enabling a dedicated integration of both semantic and geometric information for object tracking.

### 3.1.2 Track-by-Detection Paradigm

The track-by-detection paradigm (Henriques et al., 2012; Javed et al., 2022) forms the foundational framework for our GOT-Edit tracker. In this paradigm, a tracker predicts a target-specific tracking model (or filters), updates it dynamically online, and employs this model to localize the target in the current frame, thereby performing tracking by detection in an online manner.

Recent trackers (Bhat et al., 2019; Mayer et al., 2022; Chen et al., 2025a) in this paradigm employ a model predictor to generate the weights $\mathbf{W}$ for the localization head of the tracker. The weights are applied to the current frame features $z_{cur}$ through convolution or matrix multiplication to produce a classification score map $p$, which highlights the target's location in the current frame at the feature resolution:

$$p = \mathbf{W} * z_{cur}. \quad (4)$$

Our GOT-Edit framework aims to adapt the $\mathbf{W}$ with the new knowledge through online model editing. As the formulation in Eq. 4 shares a similar form to the linear equation of AlphaEdit, it allows GOT-Edit with AlphaEdit-like online model editing to make the fused knowledge semantics-preserved and geometry-aware, thereby improving the generalization of the tracker.

## 3.2 GOT-Edit

By combining 2D semantic understanding with 3D geometric reasoning, the proposed GOT-Edit enables online trackers to preserve semantic knowledge while adaptively incorporating geometric cues. In the following, we first introduce the pipeline that fuses semantics and geometry for GOT, and then describe how we achieve model editing to ensure seamless cooperation between semantic and geometric modalities.

**Feature Extraction.** Given the reference frames (from previous frames) and the current frame (to be localized), we extract their semantic features (Oquab et al., 2023), $v^s_{ref} \in \mathbb{R}^{C \times H \times W}$ and $v^s_{cur} \in \mathbb{R}^{C \times H \times W}$, and geometric features (Wang et al., 2025a), $v^g_{ref} \in \mathbb{R}^{C' \times H' \times W'}$ and $v^g_{cur} \in \mathbb{R}^{C' \times H' \times W'}$. Note that two reference frames are used, but only one is shown here for brevity.

**Alignment and Fusion.** The geometric features are aligned to match the dimensionality and resolution of semantic features using a convolutional network $Align(\cdot)$ and then fused with semantic features via a gating mechanism:

$$F_{ref} = v^s_{ref} + m_{ref} \odot Align(v^g_{ref}) \quad \text{and} \quad F_{cur} = v^s_{cur} + m_{cur} \odot Align(v^g_{cur}), \tag{5}$$

where $\odot$ denotes point-wise multiplication; $m_{ref} \in [0,1]^{C \times H \times W}$ and $m_{cur} \in [0,1]^{C \times H \times W}$ are spatial gating masks predicted from the paired semantic and geometric features via a lightweight convolution and a sigmoid function, for both of the reference and current frames, respectively.

**Model Predictor.** After fusing the semantic and geometric features, they are spatially concatenated with positional encodings and fed into the model predictor, a Transformer encoder-decoder (Mayer et al., 2022; Carion et al., 2020). The encoder $T_{enc}$ performs feature interaction, i.e.,

$$(z_{ref},\, z_{cur}) = T_{enc}([F'_{ref}, F_{cur}]), \quad \text{where} \quad F'_{ref} = F_{ref} + (L_{ref} \cdot e_{fg}). \tag{6}$$

In Eq. 6, $L_{ref}$ denotes the reference labels from past predictions, which indicate the correspondence of the target coordinates to the reference frame. $\mathbf{e_{fg}}$ is a learned foreground embedding (Mayer et al., 2022), and the operator $\cdot$ denotes point-wise multiplication with broadcasting.

The resulting features from Eq. 6, together with the learned foreground embedding $\mathbf{e_{fg}}$ serving as the query, are fed into a Transformer decoder (Mayer et al., 2022; Carion et al., 2020) $T_{dec}$, which generates the weights $\Delta \in \mathbb{R}^C$ of the localization head via:

$$\Delta = T_{dec}([z_{ref}, z_{cur}], e_{fg}). \tag{7}$$

**Localization Head.** The fused features of the current frame are then passed to the updated localization head for target localization:

$$p = \Delta * z_{cur}. \tag{8}$$

It is important to note that $F'_{ref}$ in Eq. 6 provides important information to differentiate the spatial and geometric properties of the target from the background in the reference frames and can serve as few-shot examples to guide target prediction in the current frame.

**Online Model Editing.** Integrating 3D features is beneficial for GOT by enabling geometric reasoning. However, it is crucial to balance its influence with that of semantic information, as a naive fusion strategy between modalities can introduce degradation of learned knowledge, as evident in Table 5, this will be discussed in detail in the following section. In addition, semantic knowledge remains the primary factor for distinguishing the appearance of the target from similar-looking objects or distractors in the environment.

To address this, we develop a mechanism that preserves semantic knowledge while incorporating geometric cues by reformulating Eq. 8 as follows:

$$p = (\mathbf{W}_{sem} + \mathbf{\Delta}') * z_{cur}, \tag{9}$$

where $\mathbf{W}_{sem} \in \mathbb{R}^C$ denotes the semantic weights, obtained by passing semantic features through the ***semantic model predictor***. This process is analogous to those described in Eq. 6 and Eq. 7, but uses only semantic features as input. $z_{cur} \in \mathbb{R}^{C \times HW}$ represents the fused semantic-geometric features of the current frame, as defined in Eq. 6. The perturbation weights $\mathbf{\Delta}'$ complement the semantic weights with geometric information and are defined as:

$$\mathbf{\Delta}' = P_{null}\Delta, \tag{10}$$

where $\Delta$ is obtained from Eq. 7 using the ***geometry model predictor***, and $P_{null} \in \mathbb{R}^{C \times C}$ is the null-space projection matrix computed from the semantic features.

Following AlphaEdit, we apply Singular Value Decomposition (SVD) to compute the null-space projector $P_{null}$ of the input semantic features. To ensure numerical stability and acquire a well-conditioned matrix, the inputs are whitened (Kessy et al., 2018) and then regularized with Ridge regression (Hoerl & Kennard, 1970) before performing SVD. Since the null-space projector is expected to be symmetric and idempotent, we carry them out via symmetrization (Ammari et al., 2012a;b), i.e., $P_{null} \leftarrow \frac{1}{2}\big(P_{null} + P_{null}^{\top}\big)$. This step restores symmetry, mitigates numerical drift, and stabilizes model training.

Unlike AlphaEdit, which performs offline model editing by collecting all preserved knowledge as in Eq. 1, our GOT-Edit predicts both preserved weights and perturbation weights in an online manner, enabling adaptive integration of geometric knowledge into the semantic model.

**Box Regression.** A regression decoder $RegDec$ takes the semantic–geometry enriched classification score map and the current frame features as input to predict a regression score map that provides the target bounding box in image resolution:

$$d = \textit{RegDec}\left(p \cdot z_{cur}\right), \tag{11}$$

where the operator $\cdot$ denotes channel-wise broadcasting multiplication, and the regression decoder $RegDec$, as used in (Mayer et al., 2022; Chen et al., 2025a), employs four convolutional layers to produce four feature maps $\mathbf{d}$ in the *ltrb* (left, top, right, bottom) bounding box representation (Tian et al., 2019). The coordinates with the highest classification score in $\mathbf{p}$ are mapped onto the regression score map $\mathbf{d}$ for final bounding box prediction.

**Objective Function.** The training objective is identical to that of previous work (Mayer et al., 2022; Bhat et al., 2019), i.e.,

$$\mathcal{L} = \lambda_{cls} L_{cls}(\hat{p}, p) + \lambda_{giou} L_{giou}(\hat{d}, d), \tag{12}$$

where $\hat{p}$ and $\hat{d}$ are the ground-truth labels. The target classification loss $L_{cls}$ is a compound hinge loss as described in (Bhat et al., 2019), while the GIoU loss (Rezatofighi et al., 2019) $L_{giou}$ is used to supervise bounding box regression. $\lambda_{cls}$ and $\lambda_{giou}$ are scalar weights that control the contribution of each loss, and these hyperparameters are identical to those in (Mayer et al., 2022).

# 4 EXPERIMENTAL RESULTS

## 4.1 EXPERIMENTAL SETTING

**Training Data.** Like most trackers, e.g. (Mayer et al., 2022; Chen et al., 2025a; 2023; Lin et al., 2024), we adopt the training splits of LaSOT, GOT10k, TrackingNet, and COCO for model training. Since some recent trackers (Kang et al., 2025; Liang et al., 2025) include VastTrack (Peng et al., 2024) for training, we provide a variant of our tracker trained with this new dataset. The training data rigorously follows the VOT2022 (Kristan et al., 2022) challenge and GOT-10K guidelines.

**Test Data.** We use the following datasets for tracker performance evaluation:

- **AVisT** (Noman et al., 2022): Designed for testing without a training set, it encompasses 120 short and long sequences, each averaging 664 frames under adverse visibility conditions.
- **NfS** (Galoogahi et al., 2017) and **OTB** (Wu et al., 2015): Used for testing without a training set, each dataset contains 100 sequences, with an average of 534 frames per sequence.
- **GOT-10k** (Huang et al., 2019): It has 420 short sequences with an average of 149 frames per sequence, featuring non-overlapping object classes in the training and test sets.
- **LaSOT** (Fan et al., 2019) and **TrackingNet** (Fan et al., 2019): They provide training data where test classes fully overlap with training classes. LaSOT has 280 long sequences with an average of 2k frames per sequence, and TrackingNet offers 511 short sequences, averaging 471 frames each.
- **VOT2020 (Kristan et al., 2020) and VOT2022 (Kristan et al., 2022)**: These are the 2020 and 2022 editions of the Visual Object Tracking challenge (VOT-ST2020 and VOT-STb2022).

**Evaluation Metrics.** We evaluate trackers using the following metrics:

- **SUC** (success rate): The percentage of frames in which the predicted bounding box overlaps the ground truth by at least an IoU threshold or the average of all thresholds.
- **SR75**: It refers to SUC with an IoU threshold of $75\%$.
- **OP50**: The percentage of frames where the predicted and ground truth IoU exceed $50\%$.
- **Pr** (precision): It measures the percentage of frames where the predicted target center is within $T$ pixels of the ground-truth center. $T$ is set to 20 in this work.
- **NPr** (normalized precision): It is the percentage of frames where the center location error, normalized by the target's box diagonal, is less than the threshold of $0.2$.
- **AO** (average overlap): The mean IoU between the predicted and ground-truth bounding boxes.

Table 1: **Comparison with state-of-the-art methods.** Each tracker is followed by its input resolution. The term 'Base' in the column 'Training Data of Tracker' refers to trackers trained on the classical four datasets. 'Frames' denotes the number of frames a tracker uses on each frame during evaluation. '*' denotes a tracker trained solely on the specific GOT-10k set (Huang et al., 2019).

| Training-Test Class Overlap | | | | | | Low or No Overlap | | | | | Full Overlap | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Dataset | | | | | | AVisT | NfS | OTB | GOT-10k* | | LaSOT | | | TrackingNet | |
| Tracker | Semantic Feature | Geometry Feature | Training Data of Tracker | Frames | Trainable Parameters | SUC | SUC | SUC | AO | SR75 | NPr | Pr | SUC | NPr | SUC |
| **GOT-Edit-378 (Ours)** | DiNOv2-L | VGGT | Base+VastTrack | 3 | 53M | 64.5 | 71.1 | 75.0 | 80.2* | 79.8* | 84.8 | 82.9 | 75.0 | 91.0 | 86.7 |
| **GOT-Edit-378 (Ours)** | DiNOv2-L | VGGT | Base | 3 | 53M | 63.7 | 69.9 | 73.0 | | | 85.2 | 83.2 | 75.3 | 90.6 | 86.4 |
| PiVOT-378 (Chen et al., 2025a) | DiNOv2-L | - | Base | 3 | 34M | 62.2 | 68.2 | 71.2 | 76.9 | 75.5 | 84.7 | 82.1 | 73.4 | 90.0 | 85.3 |
| LoRAT-378 (Lin et al., 2024) | DiNOv2-L | - | Base | 3 | 32M | 62.0 | 66.7 | 72.0 | 77.5 | 78.1 | 84.1 | 82.0 | 75.1 | 89.7 | 85.6 |
| ToMP-378 (Chen et al., 2025a) | DiNOv2-L | - | Base | 3 | 25M | 61.5 | 67.8 | 71.0 | - | - | 83.6 | 80.8 | 72.6 | - | - |
| ToMP-378 (Reproduced) | DiNOv2-L | - | Base+VastTrack | 3 | 25M | 62.0 | 69.0 | 71.5 | 77.5 | 75.8 | 83.7 | 80.8 | 72.7 | 89.0 | 84.2 |
| MCITrack-384 (Kang et al., 2025) | Fast-iTPN-L | - | Base+VastTrack | 5 | 287M | 62.9 | 70.6 | 72.0 | 80.0 | 80.2 | 86.1 | 85.0 | 76.6 | 92.1 | 87.9 |
| ARPTrack-384 (Liang et al., 2025) | ViT-ARP-L | - | Base+VastTrack+K700 | 7 | 460M | - | - | - | 81.5 | 80.5 | 83.4 | 81.7 | 74.2 | 91.1 | 86.6 |
| SeqTrack-384 (Chen et al., 2023) | ViT-MAE-L | - | Base | 3 | 309M | 57.8 | 66.7 | - | 74.8 | 72.2 | 81.5 | 79.3 | 72.5 | 89.8 | 85.5 |
| GRM-320 (Gao et al., 2023) | ViT-MAE-L | - | Base | 3 | 308M | 54.5 | 66.9 | 68.9 | 73.4 | 70.4 | 81.2 | 77.9 | 71.4 | 88.9 | 84.0 |
| SATrack-384 (Ma et al., 2025) | SAViT | - | Base | 6 | 310M | 58.4 | 67.5 | - | 75.4 | 73.5 | 81.4 | 78.4 | 72.0 | 89.0 | 84.7 |
| DeTrack-384 (Zhou et al., 2024) | Denoising ViT | - | Base | 3 | - | 60.2 | - | - | 77.9 | 74.9 | 81.7 | 79.1 | 72.9 | - | - |
| SAMITE-1024 (Xu et al., 2025c) | SAM 2 | - | SA-1B | 7 | - | - | 69.2 | 69.9 | 78.9 | 72.5 | 83.4 | 81.4 | 74.9 | - | 84.5 |

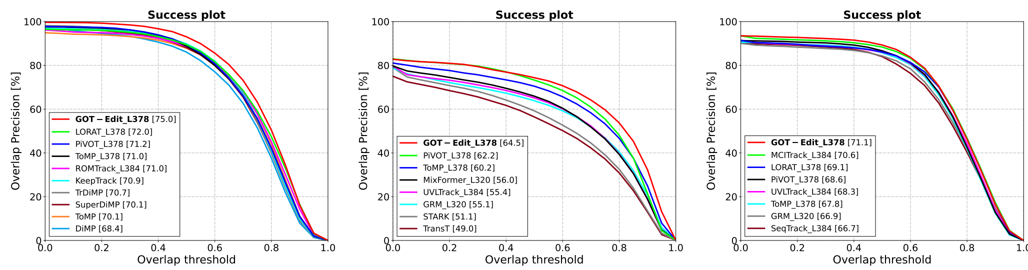


Figure 2: From left to right, success plots of competing methods on OTB, AVisT, and NfS are shown.

**Implementation Details.** Our method is implemented using PyTorch 2.0.0 and CUDA 11.7. We train the model on four A6000 GPUs (48 GB each). Inference is performed on a single NVIDIA RTX 4090 GPU and consumes approximately 9 GB of GPU memory during evaluation.

Following PiVOT (Chen et al., 2025a) and LoRAT (Lin et al., 2024), we utilize ViT-L as the backbone for image feature extraction, using weights pretrained with DINOv2 (Oquab et al., 2023). The backbone remains frozen during tracker training. For integrating geometric information, we extract intermediate features from the DPT head of VGGT (Wang et al., 2025a), which is kept frozen during training. Similar to PiVOT (Chen et al., 2025a), the model predictors and the localization head of our tracker are initialized with weights from ToMP-L, a DINOv2-L variant of ToMP. For an efficient design, the dual model predictors share the same architecture and weights, but two independent lightweight convolutional layers are appended in parallel to the predictors, serving as task-specific heads for semantic weight prediction and perturbation weight prediction, respectively.

We sample 200K subsequences per epoch and train for 30 epochs. Each subsequence consists of two reference frames and one current frame, randomly selected from a 200-frame window within a video sequence. Following ToMP (Mayer et al., 2022), PiVOT (Chen et al., 2025a), we set the search area scale factor to 5.0 and perform data augmentation. The initial learning rate is set to $10^{-4}$ with a StepLR scheduler that decays it by a factor of 0.2 at epochs 10, 15, and 20. AdamW (Loshchilov & Hutter, 2019) is used as the optimizer.

To mitigate the computational cost of higher image resolutions, as in recent works (Xie et al., 2025; Li et al., 2025; Chen et al., 2023; Lin et al., 2024), we use smaller resolutions for most ablations and higher resolutions for comparison with the state of the art: 1) **GOT-Edit-252**, where the frame resolution and the patch token size are $252 \times 252$ and $18 \times 18$, respectively; 2) **GOT-Edit-378**, where the frame resolution and the token size are $378 \times 378$ and $27 \times 27$, respectively. We also employ mixed-precision training with BFloat16 and Float32 for efficiency.

### 4.2 Comparisons with the State-of-the-Art Methods

Table 1 compares our GOT-Edit with the SOTAs on several benchmark datasets. When compared with trackers that use semantic backbones based on DINOv2 (Oquab et al., 2023), our tracker demonstrates superior performance, generalizes well to out-of-distribution targets, and achieves competitive results on in-distribution targets.

Table 2: Comparisons among trackers on the VOT challenge using Robustness as the metric.

| | GOT-Edit | PiVOT | MixFormerL | OSTrackSTB | TransT_M | ToMP |
|---|---|---|---|---|---|---|
| VOT-STb2022 | 89.8 | 87.3 | 85.9 | 86.7 | 84.9 | 81.8 |
| VOT-ST2020 | 90.3 | – | 85.5 | – | – | 78.9 |

Table 3: Comparison with trackers using DINO features under OP50.

| Dataset | AVisT | NfS | LaSOT |
|---|---|---|---|
| Tracker / Metric | | OP50 | |
| GOT-Edit-378_Vast | 74.4 | 89.3 | 85.9 |
| GOT-Edit-378 | 73.7 | 88.7 | 86.4 |
| ToMP-378 | 72.6 | 85.7 | 84.8 |
| LoRAT-378 | 72.4 | 85.6 | 85.1 |

Table 4: Ablation studies on GOT-Edit with several design choices compared across multiple datasets under SUC.

| | Semantic (DINO) | Semantic (VGGT's DINO) | Geometry (VGGT) | Null Space Constrain | Regularization | AVisT | NfS | LaSOT |
|---|---|---|---|---|---|---|---|---|
| (1) | ✓ | | | | | 59.2 | 68.5 | 70.7 |
| (2) | | | ✓ | | | 55.8 | 66.3 | 67.6 |
| (3) | | ✓ | ✓ | | | 59.9 | 67.5 | 70.9 |
| (4) | ✓ | | ✓ | | | 60.2 | 68.5 | 71.3 |
| (5) | ✓ | | ✓ | ✓ | | 61.5 | 69.3 | 72.7 |
| (6) | ✓ | | ✓ | ✓ | ✓ | 62.0 | 70.2 | 73.8 |

GOT-Edit shows a performance gain of about 2–3% across datasets compared with ToMP-378, which is a DINOv2 variant of ToMP (Mayer et al., 2022) and serves as the baseline tracker. Comparing against trackers that employ different semantic backbones, our tracker outperforms all trackers on out-of-distribution targets, except MCITrack-384 (Kang et al., 2025) on in-distribution targets, which uses a different semantic backbone and involves more trainable parameters and frames during training and evaluation. In addition to SUC, NPr, and Pr, we compare trackers using OP50 (Table 3), where all trackers share the same semantic backbone. Our tracker outperforms all others by a clear margin in this metric. We also provide the success AUC curve in Figure 2. On OTB, our method consistently shows the best results. Our tracker outperforms all trackers when $T > 0.2$ on AVisT, while outperforming MCITrack when $T < 0.7$ on NfS. Additionally, we provide an evaluation on the VOT challenge in Table 2.

### 4.3 Ablation Studies

Table 4 presents ablation studies on each GOT-Edit component under image resolution 252, trained using four classical datasets. Row (1) shows the baseline method trained using only semantic features. Row (2) shows that the GOT tracker takes features from the DPT head of VGGT. Even though these features are used to finetune the tracker with GOT data, performance still drops dramatically due to the limited discriminative ability of the geometric information. Row (3) shows the fusion of semantic features from VGGT's DINO head and geometric features from VGGT's DPT head, which yields a moderate improvement compared with using geometric features alone. Row (4) shows semantic features extracted from an independent DINO backbone, which perform better than semantic features from the DINO head of VGGT. This is because VGGT fine-tunes its DINO backbone with large-scale 3D data, distorting the original semantic representations of the DINO backbone. Row (5) shows semantic–geometry fusion under the null-space constraint, which improves performance compared with fusion without the constraint. Row (6) shows that whitening and regularization applied to input features before SVD further improve overall performance.

Overall, our online model editing strategy for geometry–semantics combination improves the baseline by an average of $2.5\%$, while the null space constraint with regularization effectively enhances fusion, yielding notable gains across datasets: $1.8\%$ on AVisT, $1.7\%$ on NfS, and $2.5\%$ on LaSOT. These results demonstrate the superiority of GOT-Edit.

Our method freezes semantic and geometry feature extractors and fuses them using the proposed knowledge-editing approach during training, enabling seamless cooperation between the two modalities and further complements the semantic distortion in VGGT, where semantic features tend to be dominated by geometry, and complements existing GOT trackers, which lack geometric knowledge.

Table 5 shows the ablation studies of GOT-Edit-252 components with regard to attributes. Row (1) presents the baseline performance. Row (2) reports the results of incorporating semantic and geometric information under a naive fusion method. For attributes related to 3D (e.g., occlusion, visibility, background clutter), the performance improves. However, for non-3D-related attributes (e.g., distractor, fast motion, illumination), the performance degrades. By addressing the fusion balancing problem through the null-space constraint, as adopted in our GOT-Edit, the tracker achieves not only geometric benefits but also semantic consistency, as demonstrated in row (3).

Table 5: Ablation studies of GOT-Edit components with regard to the attributes.

| | Semantic (DINO) | Geometry (VGGT) | Null Space Constrain | AVisT Weather Conditions (Target Visibility) | AVisT Obstruction Effects (Occlusion) | AVisT Camouflage (Background Clutter) | AVisT Target Effects (Distractor) | LaSOT Partial Occlusion | LaSOT Full Occlusion | LaSOT Background Clutter | LaSOT Fast Motion | LaSOT Illumination |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| (1) | ✓ | | | 64.32 | 57.14 | 42.21 | 49.38 | 68.97 | 62.93 | 64.25 | 60.39 | 72.02 |
| (2) | ✓ | ✓ | | 66.58 | 59.83 | 44.37 | 47.18 | 70.08 | 63.74 | 65.45 | 58.73 | 71.13 |
| (3) | ✓ | ✓ | ✓ | 67.95 | 62.67 | 46.93 | 50.27 | 71.60 | 66.33 | 67.85 | 62.90 | 73.23 |

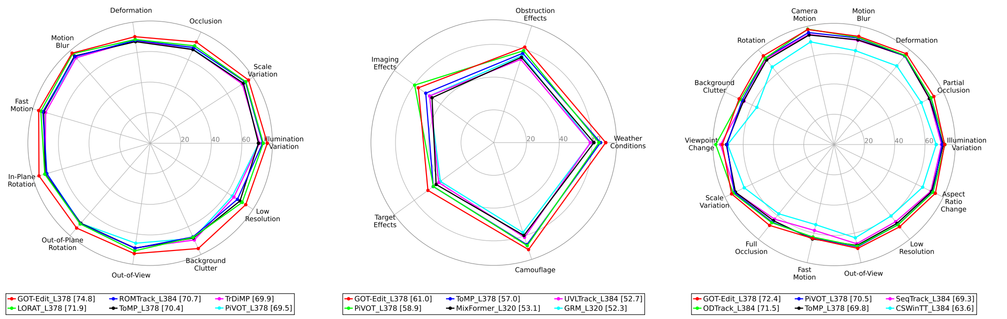


Figure 3: Attribute analysis of OTB, AVisT, and LaSOT from left to right, with average scores below.

## 4.4 Comparison of attributes among SoTA

We conduct an attribute-based analysis by comparing our GOT-Edit with several trackers like Song et al. (2022); Zheng et al. (2024); Ma et al. (2024); Cui et al. (2022); Wang et al. (2021) using large resolution input, as shown in Figure 3. This analysis provides insights into the strengths and weaknesses of different methods and highlights potential areas for improvement. Note that attribute-based plotting requires the raw results of a tracker. If the raw data of a tracker is unavailable or if datasets lack an attribute analysis protocol (e.g., those hosted on third-party servers without attribute results), we exclude those trackers from the attribute analysis.

**OTB**: As the left column of Figure 3 illustrates, our tracker achieves a considerable performance gain on attributes such as background clutter, occlusion, and rotation, compared with the baseline ToMP-L378. These improvements result from the geometry information that aids the understanding of the scene and the object itself, while other attributes still outperform competing trackers.

**AVisT**: As shown in the middle column of Figure 3, our tracker achieves improvements across most attributes compared with other trackers. Although it falls behind PiVOT in Imaging Effects (low-light images), it still outperforms the baseline ToMP-L378 across attributes, demonstrating its effectiveness in handling unseen data.

**LaSOT**: The right column of Figure 3 demonstrates that our tracker outperforms most attributes compared with other trackers; however, in viewpoint change and fast motion, it performs similarly or slightly drops below some trackers. This is because visual geometry becomes less effective when the scene or object moves rapidly or undergoes significant viewpoint changes.

**Limitations.** While improved in most attributes, our tracker still requires enhancement in handling moving objects and scenes, as evidenced in the LaSOT benchmark. The 'Target Effects' attribute in the AVisT benchmark, which contains both distractors and fast-moving objects, also provides evidence for improvement. Additionally, handling out-of-distribution data, as in AVisT, presents opportunities for further advancement.

# 5 Conclusion

We present GOT-Edit, the first framework to embed geometry-grounded reasoning into generic object tracking via online model editing without explicit 3D inputs. By constraining updates to preserve learned semantics, GOT-Edit prevents degradation while incorporating geometric cues overlooked by conventional 2D trackers. Through online model editing with null-space constraint, it retains semantic knowledge while adaptively integrating geometric information, achieving robustness under occlusion, clutter, and visual ambiguity. The framework generalizes across datasets, targets, and environments while maintaining stability and robustness. Beyond surpassing state-of-the-art trackers in generalization, the results demonstrate that principled model editing can bridge modality gaps and recover geometry information missed by purely 2D approaches. These advances chart a path toward reliability, safety, and social responsibility in vision systems.

## ETHICS STATEMENT

The proposed GOT-Edit framework enhances generic object tracking by integrating semantic and geometric reasoning through online model editing. This capability offers societal benefits, including improved reliability of autonomous and robotic systems and assistance in challenging visual environments. Nevertheless, enhanced tracking performance may be misused for intrusive surveillance or other applications that compromise privacy and security. We emphasize that deployment of this technology must comply with legal and ethical standards, particularly in contexts involving personal data or sensitive environments. The tracker training data are solely drawn from public datasets, consistent with existing methods. Responsible use requires transparency, rigorous validation, and adherence to established ethical guidelines.

## REPRODUCIBILITY

To ensure reproducibility, detailed implementation instructions for GOT-Edit are provided in 4.1. The source code will be publicly available upon acceptance. These measures are intended to facilitate the verification and replication of the results by other researchers.

## REFERENCES

Habib Ammari, Josselin Garnier, Hyeonbae Kang, Mikyoung Lim, and Knut Sølna. Multistatic imaging of extended targets. *SIAM Journal on Imaging Sciences (SIIMS)*, 5(2):564–600, 2012a.

Habib Ammari, Josselin Garnier, and Knut Sølna. A statistical approach to target detection and localization in the presence of noise. *Waves in Random and Complex Media (WRCM)*, 22(1): 40–65, 2012b.

Yifan Bai, Zeyang Zhao, Yihong Gong, and Xing Wei. Artrackv2: Prompting autoregressive tracker where to look and how to describe. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024.

Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016.

Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019.

David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2010.

Wenrui Cai, Qingjie Liu, and Yunhong Wang. Hiptrack: Visual tracking with historical prompts. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024.

Yidong Cai, Jie Liu, Jie Tang, and Gangshan Wu. Robust object modeling for visual tracking. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023.

Bing Cao, Junliang Guo, Pengfei Zhu, and Qinghua Hu. Bi-directional adapter for multimodal tracking. In *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2024.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020.

Shih-Fang Chen, Jun-Cheng Chen, I-Hong Jhuo, and Yen-Yu Lin. Improving visual object tracking through visual prompting. *IEEE Transactions on Multimedia (TMM)*, 2025a.

Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021.

Xin Chen, Houwen Peng, Dong Wang, Huchuan Lu, and Han Hu. Seqtrack: Sequence to sequence learning for visual object tracking. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023.

Xin Chen, Ben Kang, Wanting Geng, Jiawen Zhu, Yi Liu, Dong Wang, and Huchuan Lu. Sutrack: Towards simple and unified single object tracking. In *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2025b.

Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. Mixformer: End-to-end tracking with iterative mixed attention. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022.

Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016.

Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017.

Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens, Lucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang. Tap-vid: A benchmark for tracking any point in a video. In *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2022.

Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023.

Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Shi Jie, Xiang Wang, Xiangnan He, and Tat-Seng Chua. Alphaedit: Null-space constrained knowledge editing for language models. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2025.

Xiaokun Feng, Dailing Zhang, Shiyu Hu, Xuchen Li, Meiqi Wu, Jing Zhang, Xiaotang Chen, and Kaiqi Huang. Cstrack: Enhancing rgb-x tracking via compact spatiotemporal features. In *Proc. Int. Conf. Mach. Learn. (ICML)*, 2025.

Hamed Kiani Galoogahi, Ashton Fagg, Chen Huang, Deva Ramanan, and Simon Lucey. Need for speed: A benchmark for higher frame rate object tracking. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2017.

Shenyuan Gao, Chunluan Zhou, Chao Ma, Xinggang Wang, and Junsong Yuan. Aiatrack: Attention in attention for transformer visual tracking. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022.

Shenyuan Gao, Chunluan Zhou, and Jun Zhang. Generalized relation modeling for transformer tracking. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023.

Richard L Gregory. Knowledge in perception and illusion. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences (PHILOS T R SOC B)*, 1997.

Dongyan Guo, Jun Wang, Ying Cui, Zhenhua Wang, and Shengyong Chen. Siamcar: Siamese fully convolutional classification and regression for visual tracking. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020.

Mingzhe Guo, Zhipeng Zhang, Heng Fan, Liping Jing, Yilin Lyu, Bing Li, and Weiming Hu. Learning target-aware representation for visual tracking via informative interactions. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022.

Mingzhe Guo, Weiping Tan, Wenyu Ran, Liping Jing, and Zhipeng Zhang. Dreamtrack: Dreaming the future for multimodal visual object tracking. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2025.

Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022.

Adam W Harley, Yang You, Xinglong Sun, Yang Zheng, Nikhil Raghuraman, Yunqi Gu, Sheldon Liang, Wen-Hsuan Chu, Achal Dave, Pavel Tokmakov, et al. Alltracker: Efficient dense point tracking at high resolution. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2025.

Kaijie He, Canlong Zhang, Sheng Xie, Zhixin Li, and Zhiwen Wang. Target-aware tracking with long-term context attention. In *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2023.

Joao F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012.

Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics (Technometrics)*, 12(1):55–67, 1970.

Xiaojun Hou, Jiazheng Xing, Yijie Qian, Yaowei Guo, Shuo Xin, Junhao Chen, Kai Tang, Mengmeng Wang, Zhengkai Jiang, Liang Liu, et al. Sdstrack: Self-distillation symmetric adapter learning for multi-modal visual object tracking. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024.

Xiantao Hu, Ying Tai, Xu Zhao, Chen Zhao, Zhenyu Zhang, Jun Li, Bineng Zhong, and Jian Yang. Exploiting multimodal spatial-temporal patterns for video object tracking. In *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2025.

Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 2019.

Sajid Javed, Martin Danelljan, Fahad Shahbaz Khan, Muhammad Haris Khan, and Jiri Matas. Visual object tracking with discriminative filters and siamese networks: a survey and outlook. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 2022.

Xie Jinxia, Zhong Bineng, Mo Zhiyi, Zhang Shengping, Shi Liangtao, Song Shuxiang, and Ji Rongrong. Autoregressive queries for adaptive tracking with spatio-temporal transformers. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024.

Ben Kang, Xin Chen, Simiao Lai, Yang Liu, Yi Liu, and Dong Wang. Exploring enhanced contextual information for video-level object tracking. In *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2025.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2024.

Agnan Kessy, Alex Lewin, and Korbinian Strimmer. Optimal whitening and decorrelation. *The American Statistician (Am. Stat.)*, 72(4):309–314, 2018.

Hamed Kiani Galoogahi, Ashton Fagg, and Simon Lucey. Learning background-aware correlation filters for visual tracking. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2017.

Erin Koch, Famya Baig, and Qasim Zaidi. Picture perception reveals mental geometry of 3d scene inferences. *Proceedings of the National Academy of Sciences of the United States of America (PNAS)*, 2018.

Matej Kristan, Aleš Leonardis, Jiří Matas, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kämäräinen, Martin Danelljan, Luka Čehovin Zajc, Alan Lukežič, Ondrej Drbohlav, et al. The eighth visual object tracking vot2020 challenge results. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020.

Matej Kristan, Aleš Leonardis, Jiří Matas, Michael Felsberg, Martin Danelljan, Alan Lukežič, et al. The tenth visual object tracking vot2022 challenge results. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022.

Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018.

Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019.

Xiaohai Li, Bineng Zhong, Qihua Liang, Guorong Li, Zhiyi Mo, and Shuxiang Song. Mambalct: Boosting tracking via long-term context state space model. In *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2025.

Xin Li, Yuqing Huang, Zhenyu He, Yaowei Wang, Huchuan Lu, and Ming-Hsuan Yang. Citetracker: Correlating image and text for visual tracking. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023.

Shiyi Liang, Yifan Bai, Yihong Gong, and Xing Wei. Autoregressive sequential pretraining for visual tracking. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2025.

Liting Lin, Heng Fan, Zhipeng Zhang, Yaowei Wang, Yong Xu, and Haibin Ling. Tracking meets lora: Faster training, larger model, stronger performance. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2024.

Jiaming Liu, Yue Wu, Maoguo Gong, Qiguang Miao, Wenping Ma, Cai Xu, and Can Qin. M3sot: Multi-frame, multi-field, multi-space 3d single object tracking. In *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019.

Alan Lukezic, Tomas Vojir, Luka ˇCehovin Zajc, Jiri Matas, and Matej Kristan. Discriminative correlation filter with channel and spatial reliability. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017.

Yinchao Ma, Yuyang Tang, Wenfei Yang, Tianzhu Zhang, Jinpeng Zhang, and Mengxue Kang. Unifying visual and vision-language tracking via contrastive learning. In *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2024.

Yinchao Ma, Qianjin Yu, Wenfei Yang, Tianzhu Zhang, and Jinpeng Zhang. Learning discriminative features for visual tracking via scenario decoupling. *Int. J. Comput. Vis. (IJCV)*, 2025.

Christoph Mayer, Martin Danelljan, Danda Pani Paudel, and Luc Van Gool. Learning target candidate association to keep track of what not to track. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021.

Christoph Mayer, Martin Danelljan, Goutam Bhat, Matthieu Paul, Danda Pani Paudel, Fisher Yu, and Luc Van Gool. Transforming model prediction for tracking. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022.

Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018.

Ke Nai and Shaomiao Chen. Learning a novel ensemble tracker for robust visual tracking. *IEEE Trans. Multimedia (TMM)*, 2023.

Jiahao Nie, Zhiwei He, Xudong Lv, Xueyi Zhou, Dong-Kyu Chae, and Fei Xie. Towards category unification of 3d single object tracking on point clouds. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024.

Mubashir Noman, Wafa Al Ghallabi, Daniya Najiha, Christoph Mayer, Akshay Dudhane, Martin Danelljan, Hisham Cholakkal, Salman Khan, Luc Van Gool, and Fahad Shahbaz Khan. Avist: A benchmark for visual object tracking in adverse visibility. In *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2022.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Trans. Mach. Learn. Res. (TMLR)*, 2023.

Liang Peng, Junyuan Gao, Xinran Liu, Weihong Li, Shaohua Dong, Zhipeng Zhang, Heng Fan, and Libo Zhang. Vasttrack: Vast category visual object tracking. In *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2024.

Frano Rajič, Haofei Xu, Marko Mihajlovic, Siyuan Li, Irem Demir, Emircan Gündoğdu, Lei Ke, Sergey Prokudin, Marc Pollefeys, and Siyu Tang. Multi-view 3d point tracking. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2025.

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019.

Liangtao Shi, Bineng Zhong, Qihua Liang, Ning Li, Shengping Zhang, and Xianxian Li. Explicit visual prompts for visual object tracking. In *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2024.

Zikai Song, Junqing Yu, Yi-Ping Phoebe Chen, and Wei Yang. Transformer tracking with cyclic shifting window attention. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022.

Zikai Song, Run Luo, Junqing Yu, Yi-Ping Phoebe Chen, and Wei Yang. Compact transformer tracker with correlative masked modeling. In *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2023.

Yuedong Tan, Jiawei Shao, Eduard Zamfir, Ruanjun Li, Zhaochong An, Chao Ma, Danda Paudel, Luc Van Gool, Radu Timofte, and Zongwei Wu. What you have is what you track: Adaptive and robust multimodal tracking. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2025a.

Yuedong Tan, Zongwei Wu, Yuqian Fu, Zhuyun Zhou, Guolei Sun, Eduard Zamfi, Chao Ma, Danda Pani Paudel, Luc Van Gool, and Radu Timofte. Xtrack: Multimodal training boosts rgb-x video object trackers. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2025b.

Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019.

Paul Voigtlaender, Jonathon Luiten, Philip H.S. Torr, and Bastian Leibe. Siam r-cnn: Visual tracking by re-detection. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2025a.

Ning Wang, Wengang Zhou, Jie Wang, and Houqiang Li. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021.

Qianqian Wang, Yen-Yu Chang, Ruojin Cai, Zhengqi Li, Bharath Hariharan, Aleksander Holynski, and Noah Snavely. Tracking everything everywhere all at once. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 10510–10522, 2025b.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024.

Qiangqiang Wu, Tianyu Yang, Ziquan Liu, Baoyuan Wu, Ying Shan, and Antoni B Chan. Dropmae: Masked autoencoders with spatial-attention dropout for tracking tasks. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023.

Qiao Wu, Kun Sun, Pei An, Mathieu Salzmann, Yanning Zhang, and Jiaqi Yang. 3d single-object tracking in point clouds with high temporal variation. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2024.

Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 2015.

Yuxi Xiao, Jianyuan Wang, Nan Xue, Nikita Karaev, Yuri Makarov, Bingyi Kang, Xing Zhu, Hujun Bao, Yujun Shen, and Xiaowei Zhou. Spatialtrackerv2: 3d point tracking made easy. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2025.

Fei Xie, Zhongdao Wang, and Chao Ma. Diffusiontrack: Point set diffusion model for visual object tracking. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024.

Jinxia Xie, Bineng Zhong, Qihua Liang, Ning Li, Zhiyi Mo, and Shuxiang Song. Robust tracking via mamba-based context-aware token learning. In *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2025.

Chenlong Xu, Bineng Zhong, Qihua Liang, Yaozong Zheng, Guorong Li, and Shuxiang Song. Less is more: Token context-aware learning for object tracking. In *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2025a.

Mengjie Xu, Yitao Zhu, Haotian Jiang, Jiaming Li, Zhenrong Shen, Sheng Wang, Haolin Huang, Xinyu Wang, Han Zhang, Qing Yang, et al. Mitracker: Multi-view integration for visual object tracking. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2025b.

Qianxiong Xu, Lanyun Zhu, Chenxi Liu, Guosheng Lin, Cheng Long, Ziyue Li, and Rui Zhao. Samite: Position prompted sam2 with calibrated memory for visual object tracking. *arXiv preprint arXiv:2507.21732*, 2025c.

Yinda Xu, Zeyu Wang, Zuoxin Li, Ye Yuan, and Gang Yu. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2020.

Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021a.

Song Yan, Jinyu Yang, Jani Käpylä, Feng Zheng, Aleš Leonardis, and Joni-Kristian Kämäräinen. Depthtrack: Unveiling the power of rgbd tracking. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021b.

Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 21924–21935, 2025.

Jinyu Yang, Zhongqun Zhang, Zhe Li, Hyung Jin Chang, Aleš Leonardis, and Feng Zheng. Towards generic 3d tracking in rgbd videos: Benchmark and baseline. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022.

Yingjie Yao, Xiaohe Wu, Shiguang Shan, and Wangmeng Zuo. Joint representation and truncated inference learning for correlation filter based tracking. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018.

Botao Ye, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Joint feature learning and relation modeling for tracking: A one-stream framework. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022.

Yuechen Yu, Yilei Xiong, Weilin Huang, and Matthew R Scott. Deformable siamese attention networks for visual object tracking. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020.

Jingwen Zhang, Zikun Zhou, Guangming Lu, Jiandong Tian, and Wenjie Pei. Robust 3d tracking with quality-aware shape completion. In *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2024a.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. MonST3r: A simple approach for estimating geometry in the presence of motion. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2025.

Zhengbo Zhang, Li Xu, Duo Peng, Hossein Rahmani, and Jun Liu. Diff-tracker: Text-to-image diffusion models are unsupervised trackers. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2024b.

Zhipeng Zhang, Houwen Peng, Jianlong Fu, Bing Li, and Weiming Hu. Ocean: Object-aware anchor-free tracking. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020.

Haojie Zhao, Dong Wang, and Huchuan Lu. Representation learning for visual object tracking by masked appearance transfer. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023.

Yaozong Zheng, Bineng Zhong, Qihua Liang, Zhiyi Mo, Shengping Zhang, and Xianxian Li. Odtrack: Online dense temporal token learning for visual tracking. In *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2024.

Xinyu Zhou, Pinxue Guo, Lingyi Hong, Jinglun Li, Wei Zhang, Weifeng Ge, and Wenqiang Zhang. Reading relevant feature from global representation memory for visual object tracking. In *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2023.

Xinyu Zhou, Jinglun Li, Lingyi Hong, Kaixun Jiang, Pinxue Guo, Weifeng Ge, and Wenqiang Zhang. Detrack: In-model latent denoising learning for visual object tracking. In *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2024.

Jiawen Zhu, Simiao Lai, Xin Chen, Dong Wang, and Huchuan Lu. Visual prompt multi-modal tracking. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023.

Jiawen Zhu, Huayi Tang, Xin Chen, Xinying Wang, Dong Wang, and Huchuan Lu. Two-stream beats one-stream: asymmetric siamese network for efficient visual tracking. In *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2025.

# A APPENDIX

This document supplements the main paper with details on GOT-Edit, including comparisons with state-of-the-art trackers using NPr, Pr, and SUC plots, ablation on model complexity, and a **video appendix** for qualitative visualisation on LaSOT and AVisT, available as a zipped file **from the paper submission forum**.

# B COMPUTATIONAL COST ANALYSIS

Table 6: The analysis quantifies the computational costs of each component of the GOT-Edit-378 in terms of runtime per frame (milliseconds, ms) and percentage (%).

| Backbone VGGT | Backbone DINO | Align and Fuse | Model Predictors | Reg/Cls Decoders | Total |
|---|---|---|---|---|---|
| 91.9 | 23.8 | 0.7 | 4.4 | 2.3 | 123.1 |
| 75% | 18% | 1% | 4% | 2% | 100% |

The computational cost of each tracker component is reported in Tab. 6 as per-frame runtime (ms) and percentage of total time. With high-resolution inputs ($378 \times 378$), the VGGT (Wang et al., 2025a) feature extractor dominates nearly 75% of the tracker cost. The second bottleneck arises from semantic features of DINO (Oquab et al., 2023), while other components contribute negligibly compared with the dual backbones for extracting semantic and geometric features. The evaluation model precision uses BFloat16 for VGGT and Float32 for the other components.

# C MORE EXPERIMENTS

We report NPr, Pr, and SUC plots on four datasets: NfS, AVisT, LaSOT, and OTB. Other datasets, such as TrackingNet and GOT-10K, are evaluated on online servers without plots and thus excluded.

**Overview Guidelines for NPr, Pr, and Suc Plots:**

In the Precision (Pr) and Normalized Precision (NPr) plots, the x-axis denotes pixel or normalized distance thresholds, while the y-axis indicates the percentage of frames in which the distance between the predicted and ground-truth target centers falls within the specified threshold. A balance is typically sought between higher precision and lower localization error. Trackers are commonly ranked by their performance at a threshold of 0.2 in NPr or 20 pixels in Pr.

In the Success (SUC) plot, the x-axis represents the IoU thresholds (measuring the overlap between the predicted bounding box and the ground truth), while the y-axis indicates the percentage of frames in which the IoU meets or exceeds the corresponding threshold. Trackers are commonly ranked by their performance, measured as the average precision across all thresholds.

We analyze the plots for each dataset as follows:

- **NfS**: In Figure 4, our tracker outperforms others once the threshold exceeds 0.1 in NPr or 10 pixels in Pr. For SUC, it consistently surpasses all baselines across thresholds.
- **AVisT**: As shown in Figure 5, AVisT, a training-free dataset with diverse adverse scenarios. Under conditions NPr with $T < 0.3$, our tracker outperforms all baselines. For PR, our tracker outperforms competitors across thresholds. For SUC, our tracker outperforms competitors when $T > 0.4$.
- **OTB**: In Figure 6, our tracker consistently outperforms competitors e.g., (Lin et al., 2024; Chen et al., 2025a; Mayer et al., 2021; Wang et al., 2021) in SUC. For Pr and NPr, most trackers perform similarly, while our method remains significantly competitive.
- **LaSOT**: In Figure 7, on this in-distribution dataset, our tracker outperforms SOTA methods, e.g., (Zheng et al., 2024; Cai et al., 2023; Song et al., 2023; 2022) when NPr $T > 0.1$, Pr $T > 10$ pixels, and SUC $< 0.7$. LoRAT surpasses our tracker only under very strict conditions, such as NPr $T < 0.1$, Pr $T < 10$ pixels, and SUC $> 0.8$. Nevertheless, our method consistently outperforms other trackers with the same backbone, including PiVOT-L378 and ToMP-L378.

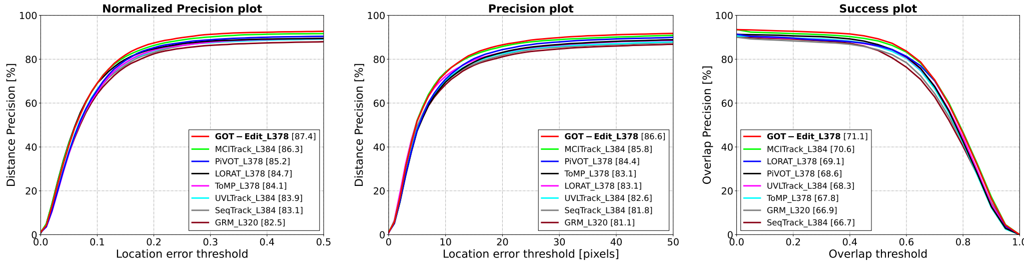


Figure 4: **Comparison of methods using NPr, Pr, and SUC on NfS, left to right.**

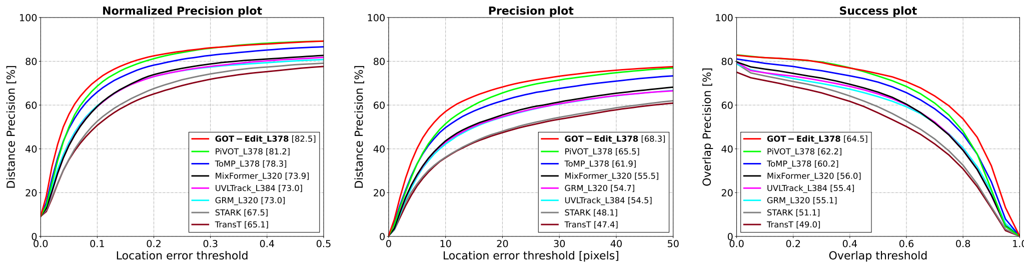


Figure 5: **Comparison of methods using NPr, Pr, and SUC on AVisT, left to right.**

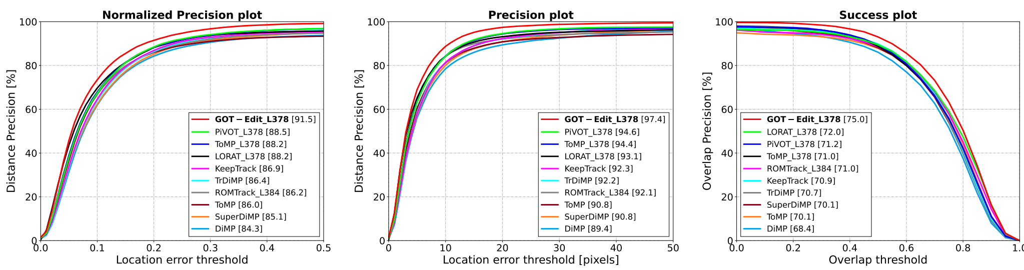


Figure 6: **Comparison of methods using NPr, Pr, and SUC on OTB, left to right.**

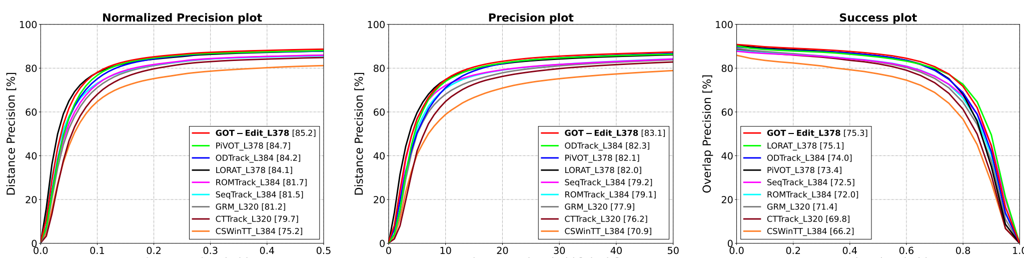


Figure 7: **Comparison of methods using NPr, Pr, and SUC on LaSOT, left to right.**

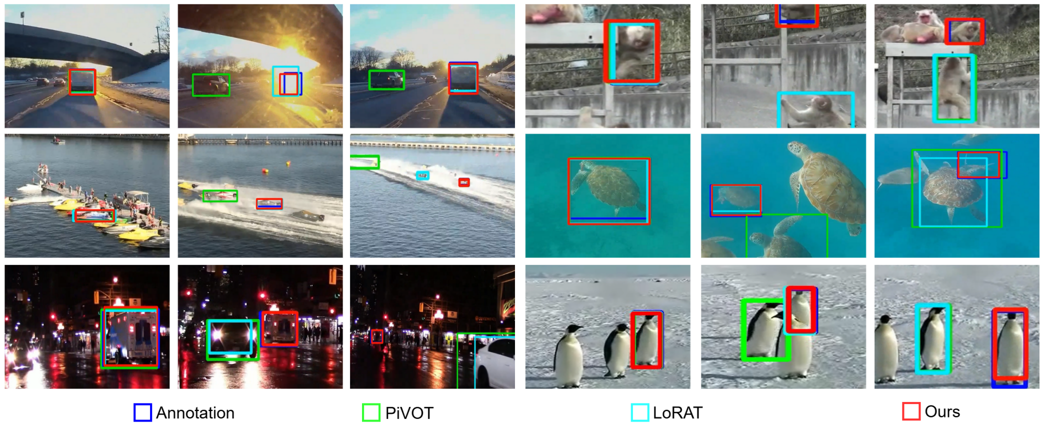


Figure 8: Visual comparisons of tracking results from GOT-Edit, PiVOT, and LoRAT across diverse video sequences under adverse scenarios are shown. The three left columns illustrate object tracking evaluation on AVisT, while the three right columns present tracking results on LaSOT. More detailed visual comparisons among the trackers are provided in the video appendix.

## D VISUALIZATION RESULTS

We present visual comparisons among trackers in Figure 8. Our tracker exhibits greater robustness under occlusion and superior discrimination against distractors. These advantages arise from the proposed method for semantic and geometric reasoning.

## E THE USE OF LARGE LANGUAGE MODELS

The research is original, and large language models were used only for polishing the writing.