This ACCV 2024 paper, provided here by the Computer Vision Foundation, is the author-created version.
The content of this paper is identical to the content of the officially published ACCV 2024
LNCS version of the paper as available on SpringerLink: https://link.springer.com/conference/accv

3D Prompt Learning for RGB-D Tracking

Bocen Li[1], Yunzhi Zhuge[1], Shan Jiang[2], Lijun Wang[1⋆], Yifan Wang[1], and Huchuan Lu[1]

[1] Dalian University of Technology, Dalian, China
[2] Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun, China

Abstract. Due to the remarkable advancements in RGB visual tracking, there has been a growing interest in RGB-D tracking, owing to its robust performance even in challenging scenarios. To bridge the gap between RGB and RGB-D tracking, several 2D prompt learning methods have emerged, which primarily target on downstream task adaptation. In contrast, we introduce a novel prompt learning method for RGB-D tracking, termed as **3D Prompt Tracking (3DPT)**, which is able to capture essential 3D geometric information and transform base RGB trackers into RGB-D trackers through parameter efficient tuning. Compared to those counterparts using depth maps as 2D prompts, we propose to directly encode 3D features from point clouds into base models, leading to more superior discriminative powers, particularly when the target and background distractors share similar visual appearance. We achieve this goal through an elaborately designed **Geometry Prompt (GP) block**, which can effectively extract 3D features, and inject the 3D knowledge into the 2D base model. The GP block is generally applicable to recent visual trackers, yielding more robust tracking performance with reasonable computational overhead. Extensive experiments demonstrate that our 3D Prompt Tracking delivers promising performance and can generalize across three popular RGB-D tracking datasets, including DepthTrack, CDTB, and VOT-RGBD2022.

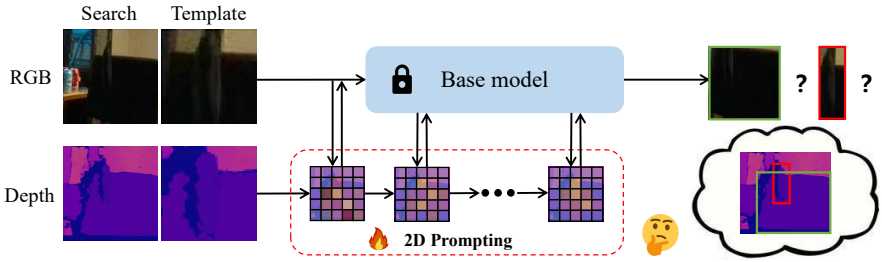
Keywords: Video object tracking · Prompt learning · Point cloud

1 Introduction

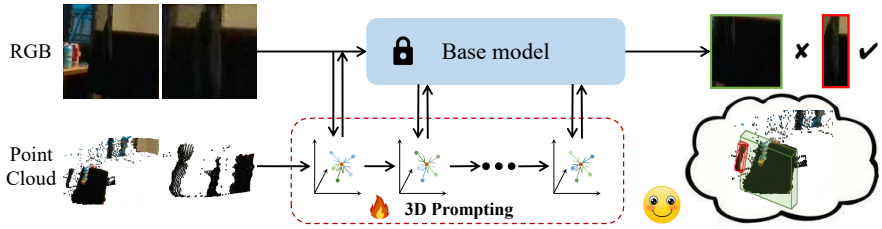
Video object tracking, as a foundational component in the field of computer vision, finds applications in various domains including virtual reality, augmented reality, and autonomous driving. Recent progress in this field is largely attributed to the adoption of transformer architecture [27] and the availability of large-scale datasets. Transformer-based RGB trackers [4, 7, 32, 33, 6, 39] have outperformed convolution-based models, benefiting from many large scale RGB tracking datasets, like LaSOT [10], GOT-10K [13], and TrackingNet [24].

Despite advancements, RGB trackers [39, 43, 8] struggle in challenging scenarios, like extreme illumination changes, background clutter, and motion blur. This

⋆ Corresponding author.



(a) 2D prompt learning



(b) 3D prompt learning

Fig. 1: Comparison between 2D prompt learning and our proposed 3D prompt learning for RGB-D tracking, where targets are marked in red and distractors in green. Point clouds are colored based on RGB images for enhanced visualization.

has led to an interest in multi-modality tracking, including RGB-Depth (RGB-D) [25, 37, 46], RGB-Thermal (RGB-T) [35, 28, 42], and RGB-Event (RGB-E) [40, 30] tracking, leveraging diverse modalities to enhance tracking performance.

Depth is a common modality with various acquisition methods [29], and has proven to be effective in many tasks [11, 45]. Therefore, we focus on RGB-D tracking in this work. Recent RGB-D trackers, which treat depth as an additional visual feature, often rely on low-level appearance cues. This can be insufficient for precise tracking in complex scenarios. Fig. 1a illustrates the limitations of depth maps in distinguishing targets from distractors, especially when they are closely located and share similar depth values.

In contrast, 3D geometric information offers a more comprehensive understanding of object shape and spatial arrangement, leading to more accurate location estimation. As shown in Fig. 1b, point clouds can clearly highlight the differences in location between the target and distractors, providing significantly more discriminative information and enhancing tracking accuracy.

While 3D information holds potential to improve tracking capabilities, effectively using it is challenging due to two main concerns. Firstly, RGB-D tracking datasets are smaller than RGB tracking datasets, posing a challenge for developing precise and robust RGB-D trackers. Secondly, integrating 3D perception into 2D base models without affecting their performance and generalization is a non-trivial challenge.

In recent years, the paradigm of prompt learning [22] has led to significant successes for neural natural language processing (NLP) algorithms in downstream tasks. This approach provides an efficient way to leverage pre-trained models for a wide range of downstream applications. Remarkably, these benefits are achieved with the addition of only a minimal number of trainable parameters.

In this paper, inspired by prompt learning, we introduce 3D Prompt Tracking (3DPT) as a solution to address the aforementioned concerns. Our approach aims to equip 2D base models with the ability to perceive the 3D environment in prompt learning manner, which results in improved performance and stronger generalization. As shown in Fig 1, different from 2D prompt learning methods, we extract and process the additional information, *i.e.* point clouds, in 3D space. Technically, we first project the depth maps into camera view. Subsequently, to extract the corresponding 3D features, we introduce a novel 3D prompt block, denoted as the Geometry Prompt (GP) block. This block treats point clouds as prompts and processes them within the 3D space. Lastly, we fuse 3D features with 2D features that come from the base model.

We apply our proposed 3D prompt tracking to two state-of-the-art RGB trackers, namely MixFormer [7] and OSTrack [39]. Experimental results demonstrate that our proposed method achieves superior performance by equipping 2D base models with 3D perceptual capabilities. In comparison to 2D prompt learning methods, our approach also exhibits enhanced performance and generalization across diverse datasets, *i.e.* DepthTrack [37], CDTB [23], and VOT-RGBD2022 datasets [16], utilizing only a limited number of trainable parameters ($1.1M$) trained on DepthTrack.

Our main contribution can be concluded as:

- We introduce a novel RGB-D tracking framework, named 3D Prompt Tracking (3DPT), that enhances 2D base models with 3D perception capabilities. This approach effectively leverages the strengths of 2D trackers pretrained on large-scale datasets while incorporating 3D geometric information with the addition of only a minimal number of parameters.
- The proposed 3D Geometry Prompt block combines 2D features and 3D representations in an efficient and effective way, which can be easily applied to other transformer-based RGB trackers, resulting in improved performance and enhanced generalization in RGB-D tracking tasks.
- Extensive results show that our proposed method achieves SOTA results on both DepthTrack and VOT-RGBD 2022 datasets, and exhibits stronger generalization across three different datasets. This achievement is expected to have a positive impact on future research endeavors involving the integration of 2D and 3D information in tracking applications.

2 Related Works

2.1 Video Object Tracking

RGB Tracking. In recent years, RGB tracking has witnessed significant advancements, largely owing to the availability of various large-scale datasets such

as LaSOT [10], Got-10K [13], and TrackingNet [24]. Additionally, the integration of transformer architectures also has made substantial contributions to the success of modern RGB trackers. For instance, TransT [5] revolutionizes video object tracking by introducing the transformer architecture, leading to a notable improvement in tracking performance. OSTrack [39] adopts the ViT [9] as its backbone, harnessing self-attention mechanisms to effectively model the relationships between search regions and templates. Stark [36], drawing inspiration from DETR [2], embraces a transformer encoder-decoder structure. SeqTrack [4] treats object tracking as a sequence generation problem and employs auto-regressive techniques to make predictions for bounding boxes. Mixformer [7] and SwinTrack [21] are two trackers that incorporate multi-scale transformer encoders into their designs.

RGB-D Tracking. In recent years, many RGB-D tracking approaches treat depth information as another form of visual information for more robust and accurate tracking. DeT [37] employs a two-encoder approach, where color and depth domain information are processed separately to extract their corresponding features. SPT [47] adopts a structure similar to Stark [36], and it feeds both color and depth images into a ResNet50 [12] backbone, followed by several transformer encoders, to extract search region and template features in both the color and depth domains.

Furthermore, building upon the impressive achievements of RGB trackers trained on extensive large-scale datasets, ProTrack [38] and ViPT [46] adopt an approach where depth information is treated as a prompt. ProTrack combines the RGB image with an additional modality into a single input image, effectively merging both sources of information. On the other hand, ViPT employs a bypass network with a few number of trainable parameters to process the additional modality. However, despite the significant progress made by recent RGB-D trackers, they still tend to treat depth maps as another form of visual information, and do not fully capitalize on the potential of modeling the geometric relationships between search regions and templates.

2.2 Prompt Learning

Recently, prompt tuning paradigm [22] has gained significant attention as an alternative to traditional full fine-tuning. Noted for its superior performance and parameter efficiency, this approach is increasingly preferred in a variety of downstream tasks.

VPT [14] utilizes predefined parameters as prompts and fine-tunes them on downstream tasks. This approach outperforms full fine-tuning methods in many downstream tasks, highlighting the potential of prompt learning in computer vision. In contrast to VPT, AdaptFormer [3] introduces an AdaptMLP module and incorporates it after the multi-head self-attention within a transformer block. ConvPass [15] utilizes a convolutional bypass subnetwork as adaptation modules in Vision Transformers (ViT) [9] to improve their performance on visual tasks, especially in low-data scenarios. ControlNet [41] introduces various

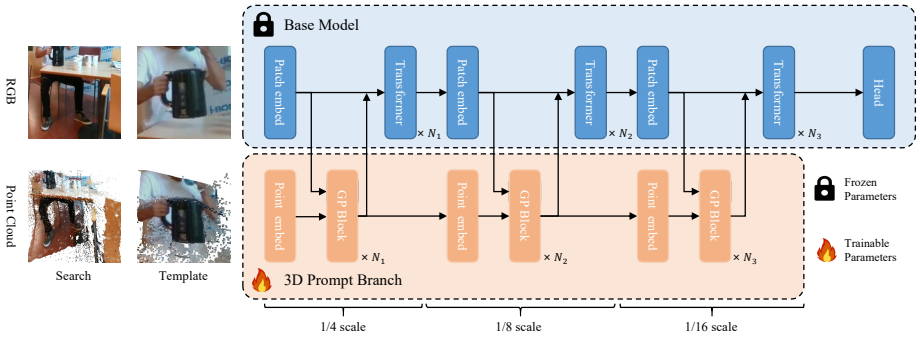


Fig. 2: The general architecture of our proposed 3D prompt tracking. The input RGB images are sent to the base model MixFormer, while the point clouds are sent to the 3D prompt branch, consisting of multiple our proposed Geometry Prompt (GP) block. At each scale, we downsample the point cloud features and their corresponding 3D coordinates using a point embedding process.

types of information, such as depth maps, Canny edges, and scribbles, which can be viewed as different types of prompts to control predictions of the diffusion model [26]. In essence, prompts serve as additional information to enhance the knowledge of base models.

Based on that, ViPT and ProTrack uses different modality information to adapt the base model to different tasks. Comparing with these 2D prompt learning method, our method focuses on equipping 2D base models with the ability to process 3D information within the prompt learning paradigm.

3 Method

In this section, we introduce our proposed method, 3D Prompt Tracking (3DPT), with the aim of equipping 2D base models with the capability to perceive and understand the 3D environment through parameter efficient tuning, harnessing the combined power of the 2D and 3D domains to significantly improve accuracy and robustness in video object tracking.

We will begin by introducing the formulation of RGB-D tracking and its application within the prompt learning paradigm. Subsequently, we provide a detailed explanation of our approach. Finally, we describe the experimental configurations for different base models.

3.1 Problem Formulation

Given a tracking dataset $\mathcal{I} = \{(I_i, B_i)\}_{i=1}^{N}$, tracking algorithms utilize the first frame image I_1 and its corresponding bounding box B_1 to predict the object location B_t in the t-th frame. This can be formally expressed as:

$$B_t = \mathcal{F}_{\theta_1}(I_t, I_1, B_1), \tag{1}$$

where $\mathcal{F}(\cdot)$ is the forward function of a base model with its corresponding parameters θ_1 trained on large scale RGB tracking datasets.

The conventional fine-tuning paradigm aims to train the parameters θ_1 on the target dataset. In RGB-D tracking, the scale of datasets is much smaller than the RGB counterpart. Directly using the full fine-tuning paradigm may lead to sub-optimal performance, as shown in Tab. 2.

To tackle this challenge, we adopt the prompt learning paradigm. In prompt learning, the goal is to introduce new knowledge into a pretrained model by adding a small number of trainable parameters θ_2 while keeping the rest parameters θ_1 frozen. In this paper, our method not only adapts the RGB base model to the RGB-D domain, but also enhances 2D models with 3D perceptual abilities. We use point clouds as prompts, allowing the model to gain 3D geometric understanding, processed by the newly added 3D prompt branch represented by θ_2. This procedure can be formulated as:

$$B_t = \mathcal{F}_{\theta_1,\theta_2}(I_t, P_t, I_1, P_1, B_1), \tag{2}$$

where P_t is the point clouds of t-th frame, which can be acquired by projecting the depth maps into 3D space.

3.2 3D Prompt Tracking

In this section, we present the details of our proposed 3D Prompt Tracking (3DPT) and its functionalities. For more concretely illustration, we choose MixFormer as the base model for demonstration, and our methods can be easily applied to other Transformer-based RGB trackers. The general structure is shown in Fig. 2. In this structure, there are two branches: the base model branch, which processes information from the RGB domain, and the 3D prompt branch, which is responsible for managing the geometric information.

Given the depth maps D_t for the t-th frame, we first project these depth maps into the 3D camera view space using the $Proj$ function:

$$P_t = Proj(D_t). \tag{3}$$

This function is based on the pinhole camera assumption, and employs camera intrinsic parameters along with translation and rotation matrices resulting from cropping, flipping, and other data augmentation techniques. For a more concise presentation, we will illustrate the process using only the template and its corresponding point cloud. It is important to note that the entire process remains identical for both the search region and the online template, and the other components are the same with the base model.

Then, the point cloud P_t is separated into different groups according to the 2D coordinates. These sub-groups of P_t are sent to a point embedding layer. The outputs from this process are regarded as point cloud prompt features.

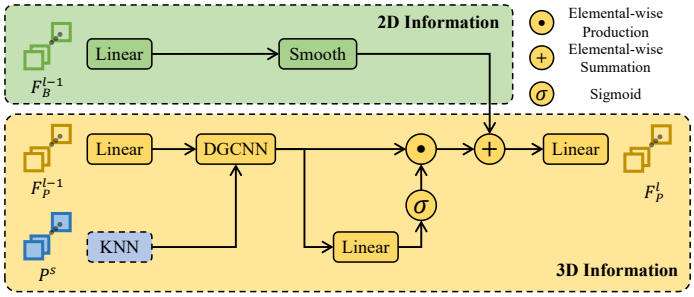


Fig. 3: The structure of the Geometry Prompt (GP) block. The purpose of this block is to merge 2D features from the base model, highlighted in green, with 3D information, highlighted in yellow. The KNN operation is executed only at the first layer of a specific scale.

Point Embedding. Due to varying resolutions within the multi-scale architecture in MixFormer, it becomes necessary to downsample the point cloud prompt features and their corresponding 3D coordinates.

The point cloud $P_t \in \mathbf{R}^{3\times H\times W}$ is generated from the depth map $D_t \in \mathbf{R}^{1\times H\times W}$, with its 2D coordinates aligning accurately with the corresponding depth map. This implies that we can directly downsample the original depth map to obtain the actual geometric information at different scales. Technically, we uniformly divide the depth map D_t^s at scale s into $\frac{H}{M}\times\frac{W}{M}$ groups, based on their 2D coordinates. Here, H and W represent the width and height, while M denotes the downsampling factor. For each group, we select the minimum depth value among all valid depth values as the downsampled depth value [20], which can be expressed as follows:

$$D_t^{s+1} = \min_{M\times M}(D_t^s \odot \mathbb{I}(D_t^s > 0)), \tag{4}$$

where $\odot$ represents element-wise production. $\mathbb{I}(\cdot)$ serves as an indicator, where it equals 1 if the condition is true and 0 otherwise. Then, we project the downsampled depth map $D_t^{s+1} \in \mathbf{R}^{1\times H_{s+1}\times W_{s+1}}$ into 3D space, resulting in the corresponding 3D point cloud $P_t^{s+1} \in \mathbf{R}^{3\times H_{s+1}W_{s+1}}$ at scale $s+1$, where $H_{s+1} = H_s/M$ and $W_{s+1} = W_s/M$. As for the point cloud prompt features, we rearrange it into 2D space. Following this, we utilize an identical downsampling structure in the corresponding base model to downsample the point cloud prompt features.

Geometry Prompt Block. For a specific scale s at the t-th frame, our objective is to integrate 2D features, originating from the base model, with 3D information. The subscripts s and t are omitted for concise illustration. In this section, we introduce our proposed Geometry Prompt (GP) block, which implements this goal in a memory and computation efficient manner. The structure is shown in Fig. 3. This block consists of two branches: one dedicated to incor-

porating semantic information from the 2D base model, while the other focuses on incorporating 3D geometric information.

Technically, the l-th block at scale s inputs the base model feature $F_B^{l-1} \in \mathbf{R}^{C_s \times H_s \times W_s}$, which is the output of the $l-1$-th transformer block, and the point cloud feature $F_P^{l-1} \in \mathbf{R}^{C_s \times H_s W_s}$, subsequently generating the prompted point cloud feature $F_P^l \in \mathbf{R}^{C_s \times H_s W_s}$.

In this block, the base model feature F_B^{l-1} is initially passed through a *Linear* layer to reduce the channel dimensions from C_s to C_h. The resulting hidden feature $F_B'^{l-1}$ with a channel dimension of C_h is then fed to a fovea smoothing block, which enhances the relevant information from the base model [46]:

$$F_P^{2D} = Fovea(F_B'^{l-1}) = softmax(\alpha F_B'^{l-1}) \odot F_B'^{l-1}, \tag{5}$$

where $softmax(x)$ denotes the softmax operation along the spatial dimension, and α is a learnable parameter initialized to 10 in our experiments, which follows the same settings in [46]. The fovea smoothing block highlights tokens with high responses while suppressing those with low responses.

Meanwhile, the point cloud feature $F_P^{l-1} \in \mathbf{R}^{C_s \times H_s W_s}$ is passed through another *Linear* layer to reduce the channel dimension to C_h. For any arbitrary token $f_i^{3D} \in \mathbf{R}^{C_h \times 1}$ within the reduced geometric feature $F_P'^{l-1}$ and its corresponding 3D coordinate $p_i \in P^s$, we select the K nearest neighbors $P_i^{KNN} \in \mathbf{R}^{K \times 3}$ in the 3D camera view space. We then model the geometric relationship between the point $p_i \in P^s$ and the point $p_j \in P_i^{KNN}$ as follows:

$$f_{i,j}^{KNN} = f_j^{3D} \ominus f_i^{3D}, i \in [1, N], j \in [1, K], \tag{6}$$

where N is the number of points, and $\ominus$ is elemental-wise minus. Subsequently, we feed both $f^{KNN} \in \mathbf{R}^{C_h \times H_s W_s \times K}$ and the original point feature $F_P'^{l-1}$ to a lightweight DGCNN (Dynamic Graph Convolutional Neural Network) [31] to capture local geometric structures. It is important to note that we perform the KNN grouping only once per scale, as the actual 3D relationships remain constant within a specific scale.

Furthermore, since the 3D points may contain unreliable points, we incorporate an additional confidence layer. This layer comprises a linear layer followed by a sigmoid operation, which serves to re-weight the geometric point feature f_{geo}^{3D}, the output of the DGCNN:

$$F_P^{3D} = \sigma(Linear(f_{geo}^{3D})) \odot f_{geo}^{3D}, \tag{7}$$

where σ denotes the sigmoid operation. The final prompted feature F_P^l is derived by rearranging $F_P^{3D} \in \mathbf{R}^{C_h \times H_s W_s}$ into 2D coordinates, then adding it to F_p^{2D}. This resultant feature is subsequently passed through another linear layer, which upsamples the channel dimension from C_h to C_s.

For the l-th transformer block, the input feature denoted as $F_{inp}^l \in \mathbf{R}^{C_s \times H_s \times W_s}$ is elemental-wise summation of two features:

$$F_{inp}^l = F_B^{l-1} \oplus F_P^l, \tag{8}$$

where F_B^{l-1} is the output of the $l-1$-th transformer block in the base model, and F_P^l is the output of the l-th geometry prompt block.

3.3 Optimization and Losses

We choose two RGB trackers as our base models: OSTrack and MixFormer. In our experiments, our goal is to predict the bounding box B_t of the t-th frame by utilizing both the base model $\mathcal{F}_{\theta_1}$ and the 3D prompt branch $\mathcal{F}_{\theta_2}$. The former aims to process 2D information, while the latter is intended for processing 3D point cloud information. During training, only the 3D prompt branch updates the parameters θ_2. Thus, the entire optimization can be formulated as follows:

$$\theta_2 = \arg\min_{\theta_2} \left[\frac{1}{|\mathcal{I}|} \sum L\left(F_{\theta_1,\theta_2}\left(I, P, B_1\right), B_{gt}\right) \right], \tag{9}$$

where $\mathcal{I}$ represents the entire dataset, encompassing RGB images I, their corresponding point clouds P, and ground truth bounding box B_{gt}.

Regarding the loss functions, different base models utilize different loss functions. For OSTrack, being a one-stage RGB tracker, it has the capability to generate both regression and prediction confidence simultaneously. Consequently, the loss functions of OSTrack [39] are consistent with the original settings:

$$L = L_{cls} + \lambda_{iou} L_{iou} + \lambda_{L_1} L_1, \tag{10}$$

where L_{cls} denotes the classification loss, while L_{iou} and L_1 represent the IoU loss and $L1$ loss, respectively.

On the other hand, MixFormer requires training the score branch at another stage. For the sake of simplicity, we only employ the losses of the first stage to train the 3D prompt branch [7]:

$$L = \lambda_{iou} L_{iou} + \lambda_{L_1} L_1, \tag{11}$$

During testing, we load the score branch checkpoint directly, without retraining.

4 Experiments

Our proposed 3D prompt tracking integrates 2D features and 3D geometric information within a unified framework. To evaluate the effectiveness of our approach, we conduct experiments on three widely-used RGB-D tracking datasets: DepthTrack [37], VOT-RGBD2022 [16], and CDTB [23]. Among these datasets, only DepthTrack provides access to its training set. In order to evaluate the precision and generalization of 3D prompt tracking, we train our models only on the training set of DepthTrack and subsequently evaluate their performance across all three datasets mentioned above.

For the base models, we select two RGB trackers, *i.e.* OSTrack and MixFormer, and initialize them using the pretrained weights of OSTrack256 and MixFormer256, respectively. These two base models are representative of different transformer architectures. MixFormer employs a multi-scale approach, which means it captures more detailed information at different scales. Conversely, OSTrack adopts a large kernel-size patch embedding and applies self-attention [27] only on the 1/16 scale. Furthermore, we also conduct experiments to investigate the influence of our proposed method on different structures in Stark, which is shown in Sec. 4.5.

Table 1: Quantitative results of different trackers on the DepthTrack test dataset, VOT-RGBD2022, and CDTB. The values ranked in the first, second, and third place are marked in **red**, **blue**, and **green**, respectively.

	Modality	DepthTrack test			VOT-RGBD2022			CDTB		
		F(↑)	Rec(↑)	Pre(↑)	EAO(↑)	Acc(↑)	Rob(↑)	F(↑)	Rec(↑)	Pre(↑)
DiMP [1]	RGB	0.436	0.418	0.456	0.543	0.703	0.731	0.570	0.570	0.570
DRefine [18]	RGBD	0.465	0.448	0.484	0.592	0.775	0.76	0.708	0.708	0.708
DMTracker [16]	RGBD	0.608	0.597	**0.619**	0.658	0.758	0.851	0.648	0.644	0.652
DAL [25]	RGBD	0.429	0.369	0.512	-	-	-	0.592	0.565	0.662
DeT	RGBD	0.532	0.506	0.560	0.657	0.760	0.845	0.657	0.642	0.674
SPT [47]	RGBD	0.538	0.549	0.527	0.651	0.798	0.851	0.688	0.726	0.654
ProTrack [38]	RGBD	0.578	0.573	0.583	0.651	0.801	0.802	**0.757**	**0.767**	**0.747**
ARKitTrack [44]	RGBD	**0.612**	**0.607**	**0.617**	0.661	0.813	0.806	0.696	0.674	**0.721**
ViPT [46]	RGBD	0.594	0.596	0.592	**0.721**	0.815	**0.871**	0.687	0.692	0.682
Un-Track [34]	RGBD	0.610	**0.610**	0.610	**0.718**	**0.820**	**0.864**	-	-	-
OSTrack [39]	RGB	0.529	0.522	0.536	0.676	0.803	0.833	**0.726**	**0.733**	**0.720**
OSTrack_3DPT (Ours)	RGBD	**0.617** (+16.3%)	**0.616** (+18.0%)	**0.619** (+15.5%)	**0.733** (+8.4%)	**0.822** (+2.4%)	**0.883** (+6.0%)	**0.725** (−0.1%)	0.736 (+0.4%)	0.714 (−0.8%)
MixFormer [7]	RGB	0.509	0.482	0.540	0.620	0.800	0.774	0.696	0.704	0.688
MixFormer_3DPT (Ours)	RGBD	**0.620** (+21.8%)	**0.610** (+26.6%)	**0.629** (+16.5%)	0.700 (+12.9%)	**0.818** (+2.2%)	0.847 (+9.4%)	0.711 (+2.2%)	0.708 (+0.6%)	0.713 (+3.6%)

4.1 Experimental Settings

We conduct our experiments on four NVIDIA 3090 GPUs with a batch size of 64 in total. The training process comprises 35 epochs, each consisting of 6×10^4 sample pairs. We employ the AdamW optimizer with a weight decay of 1×10^{-4}. The learning rate is set to 2×10^{-4} for OSTrack and 5×10^{-5} for MixFormer. The learning rate is not adjusted during the entire training process.

The evaluations are based on the official VOT protocol. For a fair comparison, all base models and our proposed method operate under short-term settings, which means templates are not updated during the tracking process.

4.2 Comparing with SOTA

The results are presented in Tab. 1. Our proposed methods outperform other recent RGB-D trackers and the base models on the DepthTrack and VOT-RGBD2022 datasets. On the CDTB dataset, our proposed methods achieve similar or better performance compared to the corresponding base models.

When compared with similar prompt-learning-based RGB-D trackers like ViPT and ProTrack, our proposed methods exhibit superior generalization across different datasets, highlighting the effectiveness of combining 2D features with 3D geometric information. Further comparisons between 2D and 3D prompt learning will be discussed in Sec. 4.4.

Additionally, ARTKitTrack [44] also extends 2D to 3D space by projecting features into a Bird's Eye View (BEV), but the projection only uses the depth map of the search region. Furthermore, BEV primarily aims to detect larger objects, such as cars and people, and the predefined BEV grid size (*e.g.* $0.2m \times 0.2m$) may not be sufficiently fine-grained to capture objects of arbitrary shapes. This limitation could result in a loss of details and potential performance degradation, particularly in unseen scenarios. In contrast, our approach leverages point clouds, offering enhanced flexibility and appropriateness for tracking

Table 2: Ablation of the main componets in the proposed method. The input of prompt branch can be categorized into three types: no input (None), depth maps (depth), and point clouds (PC). Prompt blocks (PB) are classified as no prompt (None), 2D prompt (2D), and our proposed Geometry Prompt (GP). Score Branch (SB) indicates whether the output of the score branch is utilized.

input	PB	SB	Params	F(↑)	Pre(↑)	Rec(↑)
None	None	✘	0 (0%)	0.499	0.489	0.509
None	None	✔	0 (0%)	0.503	0.522	0.486
depth	2D	✘	1.2M (3.2%)	0.583	0.572	0.594
PC	2D	✘	1.2M (3.2%)	0.605	0.592	0.618
PC	GP	✘	1.1M (3.0%)	0.615	0.602	0.628
PC	GP	✔	1.1M (3.0%)	**0.620**	**0.629**	**0.610**
PC	GP	✔	37.2M (100%)	0.586	0.593	0.579
PC	GP	✘	37.2M (100%)	0.575	0.563	0.587

objects of any shape. Experimental results demonstrate that the performance of our method not only matches ARTKitTrack on DepthTrack but also surpasses it on the other two datasets, even when ARTKitTrack employs a stronger base model setting, *i.e.* OSTrack384.

In conclusion, our method not only achieves state-of-the-art performance on both the DepthTrack and VOT-RGBD2022 datasets but also shows stronger generalization. This dual achievement underscores the effectiveness of our approach, marking a significant advancement in object tracking.

4.3 Ablation Study

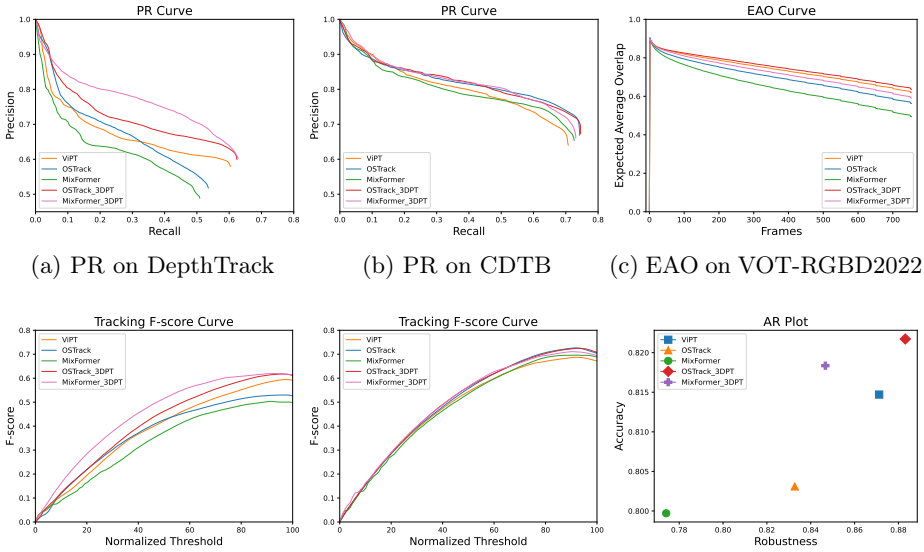
We conduct an ablation study of the Geometry Prompt block using MixFormer as the base model on the DepthTrack dataset. The model is divided into two components: transformer backbones and the score branch, with the latter not fine-tuned to focus on the impact of the backbones. In Tab. 2, the first two rows show the performance of MixFormer with and without the score branch, while the last two rows represent the fully fine-tuned version of our method.

Comparing point clouds and 2D depth maps as inputs reveals a significant advantage for point clouds. However, using point clouds alone does not fully capture 3D spatial relationships. Our Geometry Prompt (GP) block enhances the understanding of 3D geometry, enabling better discernment of spatial relationships between targets and distractors.

Additionally, our 3D prompt learning method outperforms full fine-tuning. We believe that the smaller training set for RGB-D tracking leads to catastrophic forgetting in full fine-tuning methods. In contrast, our 3DPT method, based on prompt-tuning, preserves the generalization ability of pre-trained models while enhancing their 3D perception cost-effectively.

4.4 Comparison between 2D and 3D Prompt

Quantitative Results. We compare the effectiveness of 2D and 3D prompt learning in tracking tasks on DepthTrack, CDTB, and VOT-RGBD2022 datasets.



(a) PR on DepthTrack (b) PR on CDTB (c) EAO on VOT-RGBD2022

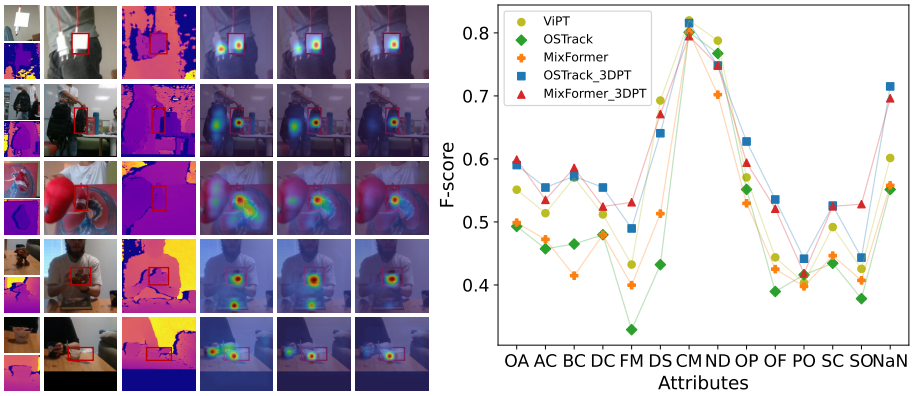
(d) F-scores on DepthTrack (e) F-scores on CDTB (f) AR on VOT-RGBD2022

Fig. 4: The metric curves on different datasets. 4a, 4b, 4d and 4e are based on VOT RGB-D 2021 format, and 4c as well as 4f are based on VOT RGB-D 2022. All quantitative results are calculated by using official VOT toolkit.

The metric curves in Fig.4 illustrate the performance of two base models alongside our 3D Prompt Tracking (3DPT) methods and the 2D prompt learning method ViPT[46]. Our 3DPT methods consistently outperform their counterparts, demonstrating significant enhancements across diverse settings and base models. In contrast, ViPT exhibits suboptimal performance and generalization. These factors show that our 3D prompt learning approach not only achieves superior performance but also shows stronger generalization, highlighting the effectiveness of 3D information integration in tracking tasks.

Visualizations. To further investigate the influence of various components in our proposed method, we provide the score maps under different settings with the base model OSTrack, which is shown in Fig. 5a. The first column is the RGB images and depth maps of different templates, and second and third columns are the RGB images and depth maps of search regions. The corresponding score maps, visualized under different experimental setups, are arranged in the last three columns. These setups include: (1) using 2D depth maps and 2D prompt blocks, (2) using point clouds and 2D prompt blocks, and (3) using point clouds alongside 3D prompt blocks (our GP blocks).

As shown in Fig. 5a, leveraging only 2D information may confuse the network, trained to detect low-level features (*e.g.* edges, contours), especially when distractors have depth values close to the target. Point clouds can offer richer



(a) Score maps (b) Attribute analysis

Fig. 5: (a) is the score maps based on the base model OSTrack. (b) are attribute F-scores on DepthTrack test dataset. We select 13 different attributes provided by the dataset, including aspect change (AC), background clutter (BC), depth change (DC), fast motion (FM), dark scene (DS), camera motion (CM), non-rigid deformation (ND), out of plane (OP), out of frame (OF), partial occlusion (PO), size change (SC), similar objects (SO) and unassigned (NaN).

and more discriminative information as depicted in Fig. 1b, mitigating this issue but still fall short in fully understanding the 3D environment. In the last column, considering that the target maintain similar 3D geometric relationship across different frames, our proposed GP blocks can model this relationship in 3D space, and enables the network to focus on the right object more accurately.

Attribute Analysis. We provide F-scores for various attributes as provided by the DepthTrack, following the approach in [19]. For each attribute, we gather all frames labeled with it into a super-sequence to calculate its F-score. This method slightly differs from the standard evaluation [17]. Hence, we provide an overall performance (OA) F-score, derived by combining all sequences into a super-sequence for F-score computation.

The results are shown in Fig. 5b. Our proposed methods outperform the base models and the 2D prompt learning method across most attributes. Notably, for scenarios that demand trackers to possess stronger discrimination capabilities, such as fast motion (FM) and similar objects (SO), MixFormer_3DPT exhibits better performance. We conjecture that multi-scale settings can provide more discriminative information than single-scale settings in such scenarios.

4.5 Generalization on the other Architecture

Beyond MixFormer and OSTrack, we also apply our proposed method to another transformer-based RGB tracker, namely Stark [36]. We evaluate the impact of

Table 3: Applying our method to Stark on DepthTrack Test set. Enc refers to the type of encoder utilized for encoding the additional modality, *i.e.* point clouds. GP denotes whether the proposed 3D geometry prompt block is employed.

Enc	GP	params	F(↑)	Pre(↑)	Rec(↑)
None	✘	None	0.469	0.519	0.428
OL	✔	1.1M(3.8%)	0.577 (+23.0%)	0.587 (+13.1%)	0.568 (+32.7%)
ML	✔	3.0M(9.6%)	0.564 (+20.3%)	0.584 (+12.5%)	0.546 (+27.6%)

GP block on different architectures by integrating it into both a transformer encoder and a ResNet image encoder, subsequently evaluating performance differences. To extract 3D geometric features, we explore two different configurations in our experiments:

- OL (OSTrack-like): Similar to OSTrack_3DPT, we embed our proposed GP block bypassing the transformer encoder, which contains only the 1/16 scale.
- ML (MixFormer-Like): Similar to MixFormer_3DPT, we embed the GP block before the first ResBlock in every scale.

The training settings remain consistent with those mentioned above. As shown in Tab. 3, when applying the proposed method to transformer encoder and ResNet both bring significant improvement comparing with the base model Stark. However, applying our proposed method to ResNet results in a slightly lower performance improvement. This suggests that the proposed Geometry Prompt (GP) block is better suited for transformer architectures. In the case of convolutional architectures like ResNet, the effective incorporation of 3D information remains an open question.

5 Conclusion

Our paper introduces a novel framework, 3D prompt tracking, which enhances 2D RGB trackers with 3D perceptual capabilities. Utilizing prompt learning techniques, this method demonstrates efficient parameter usage and significant improvement in video object tracking. Extensive experiments, which include attribute analysis and extensive testing across different datasets, consistently highlight the superiority of our 3D prompt tracking approach over traditional methods. These results reveal that our method not only achieves better performance but also exhibits robust generalization across various datasets, namely DepthTrack, CDTB and VOT-RGBD2022 respectively, even when facing some challenging scenarios. This integration of 3D perceptual abilities into existing 2D base trackers may pave the way for further exploration and progress in the domain of object tracking.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (U23A20386, 62422610, 62276045, 62293540, 62293542), Dalian Science and Technology Talent Innovation Support Plan (2022RY17, 2023JJ11CG001).

References

1. Bhat, G., Danelljan, M., Gool, L.V., Timofte, R.: Learning discriminative model prediction for tracking. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6182–6191 (2019)
2. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
3. Chen, S., Ge, C., Tong, Z., Wang, J., Song, Y., Wang, J., Luo, P.: Adaptformer: Adapting vision transformers for scalable visual recognition. Advances in Neural Information Processing Systems **35**, 16664–16678 (2022)
4. Chen, X., Peng, H., Wang, D., Lu, H., Hu, H.: Seqtrack: Sequence to sequence learning for visual object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14572–14581 (2023)
5. Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., Lu, H.: Transformer tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8126–8135 (2021)
6. Chen, Y.H., Wang, C.Y., Yang, C.Y., Chang, H.S., Lin, Y.L., Chuang, Y.Y., Liao, H.Y.M.: Neighbortrack: Improving single object tracking by bipartite matching with neighbor tracklets. arXiv preprint arXiv:2211.06663 (2022)
7. Cui, Y., Jiang, C., Wang, L., Wu, G.: Mixformer: End-to-end tracking with iterative mixed attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13608–13618 (2022)
8. Dai, K., Zhao, J., Wang, L., Wang, D., Li, J., Lu, H., Qian, X., Yang, X.: Video annotation for visual tracking via selection and refinement. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10296–10305 (2021)
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
10. Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Bai, H., Xu, Y., Liao, C., Ling, H.: Lasot: A high-quality benchmark for large-scale single object tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5374–5383 (2019)
11. He, J., Wang, Y., Wang, L., Lu, H., Luo, B., He, J.Y., Lan, J.P., Geng, Y., Xie, X.: Towards deeply unified depth-aware panoptic segmentation with bi-directional guidance learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4111–4121 (2023)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
13. Huang, L., Zhao, X., Huang, K.: Got-10k: A large high-diversity benchmark for generic object tracking in the wild. IEEE transactions on pattern analysis and machine intelligence **43**(5), 1562–1577 (2019)
14. Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., Lim, S.N.: Visual prompt tuning. In: European Conference on Computer Vision. pp. 709–727. Springer (2022)
15. Jie, S., Deng, Z.H.: Convolutional bypasses are better vision transformer adapters. arXiv preprint arXiv:2207.07039 (2022)

16. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Kämäräinen, J.K., Chang, H.J., Danelljan, M., Zajc, L.Č., Lukežič, A., et al.: The tenth visual object tracking vot2022 challenge results. In: European Conference on Computer Vision. pp. 431–460. Springer (2022)
17. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Kämäräinen, J.K., Danelljan, M., Zajc, L.Č., Lukežič, A., Drbohlav, O., et al.: The eighth visual object tracking vot2020 challenge results. In: Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16. pp. 547–601. Springer (2020)
18. Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Pflugfelder, R., Kämäräinen, J.K., Chang, H.J., Danelljan, M., Cehovin, L., Lukežič, A., et al.: The ninth visual object tracking vot2021 challenge results. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 2711–2738 (2021)
19. Kristan, M., Matas, J., Leonardis, A., Vojíř, T., Pflugfelder, R., Fernandez, G., Nebehay, G., Porikli, F., Čehovin, L.: A novel performance evaluation methodology for single-target trackers. IEEE transactions on pattern analysis and machine intelligence **38**(11), 2137–2155 (2016)
20. Li, Y., et al.: Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In: AAAI (2023)
21. Lin, L., Fan, H., Zhang, Z., Xu, Y., Ling, H.: Swintrack: A simple and strong baseline for transformer tracking. Advances in Neural Information Processing Systems **35**, 16743–16754 (2022)
22. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ACM Computing Surveys **55**(9), 1–35 (2023)
23. Lukezic, A., Kart, U., Kapyla, J., Durmush, A., Kamarainen, J.K., Matas, J., Kristan, M.: Cdtb: A color and depth visual object tracking dataset and benchmark. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10013–10022 (2019)
24. Muller, M., Bibi, A., Giancola, S., Alsubaihi, S., Ghanem, B.: Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In: Proceedings of the European conference on computer vision (ECCV). pp. 300–317 (2018)
25. Qian, Y., Yan, S., Lukežič, A., Kristan, M., Kämäräinen, J.K., Matas, J.: Dal: A deep depth-aware long-term tracker. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 7825–7832. IEEE (2021)
26. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
27. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
28. Wang, C., Xu, C., Cui, Z., Zhou, L., Zhang, T., Zhang, X., Yang, J.: Cross-modal pattern-propagation for rgb-t tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7064–7073 (2020)
29. Wang, L., Wang, Y., Wang, L., Zhan, Y., Wang, Y., Lu, H.: Can scale-consistent monocular depth be learned in a self-supervised scale-invariant manner? In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 12727–12736 (2021)
30. Wang, X., Li, J., Zhu, L., Zhang, Z., Chen, Z., Li, X., Wang, Y., Tian, Y., Wu, F.: Visevent: Reliable object tracking via collaboration of frame and event flows. arXiv preprint arXiv:2108.05015 (2021)

31. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM Transactions on Graphics (tog) **38**(5), 1–12 (2019)
32. Wei, X., Bai, Y., Zheng, Y., Shi, D., Gong, Y.: Autoregressive visual tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9697–9706 (2023)
33. Wu, Q., Yang, T., Liu, Z., Wu, B., Shan, Y., Chan, A.B.: Dropmae: Masked autoencoders with spatial-attention dropout for tracking tasks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14561–14571 (2023)
34. Wu, Z., Zheng, J., Ren, X., Vasluianu, F.A., Ma, C., Paudel, D.P., Van Gool, L., Timofte, R.: Single-model and any-modality for video object tracking. arXiv preprint arXiv:2311.15851 (2023)
35. Xiao, Y., Yang, M., Li, C., Liu, L., Tang, J.: Attribute-based progressive fusion network for rgbt tracking. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 2831–2838 (2022)
36. Yan, B., Peng, H., Fu, J., Wang, D., Lu, H.: Learning spatio-temporal transformer for visual tracking. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10448–10457 (2021)
37. Yan, S., Yang, J., Käpylä, J., Zheng, F., Leonardis, A., Kämäräinen, J.K.: Depthtrack: Unveiling the power of rgbd tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10725–10733 (2021)
38. Yang, J., Li, Z., Zheng, F., Leonardis, A., Song, J.: Prompting for multi-modal tracking. In: Proceedings of the 30th ACM International Conference on Multimedia. pp. 3492–3500 (2022)
39. Ye, B., Chang, H., Ma, B., Shan, S., Chen, X.: Joint feature learning and relation modeling for tracking: A one-stream framework. In: European Conference on Computer Vision. pp. 341–357. Springer (2022)
40. Zhang, J., Yang, X., Fu, Y., Wei, X., Yin, B., Dong, B.: Object tracking by jointly exploiting frame and event domain. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13043–13052 (2021)
41. Zhang, L., Agrawala, M.: Adding conditional control to text-to-image diffusion models. arXiv preprint arXiv:2302.05543 (2023)
42. Zhang, P., Zhao, J., Bo, C., Wang, D., Lu, H., Yang, X.: Jointly modeling motion and appearance cues for robust rgb-t tracking. IEEE Transactions on Image Processing **30**, 3335–3347 (2021)
43. Zhang, Y., Wang, L., Wang, D., Qi, J., Lu, H.: Learning regression and verification networks for robust long-term tracking. International Journal of Computer Vision **129**(9), 2536–2547 (2021)
44. Zhao, H., Chen, J., Wang, L., Lu, H.: Arkittrack: a new diverse dataset for tracking using mobile rgb-d data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5126–5135 (2023)
45. Zhou, J., Wang, L., Lu, H., Huang, K., Shi, X., Liu, B.: Mvsalnet: Multi-view augmentation for rgb-d salient object detection. In: European Conference on Computer Vision. pp. 270–287. Springer (2022)
46. Zhu, J., Lai, S., Chen, X., Wang, D., Lu, H.: Visual prompt multi-modal tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9516–9526 (2023)
47. Zhu, X.F., Xu, T., Tang, Z., Wu, Z., Liu, H., Yang, X., Wu, X.J., Kittler, J.: Rgbd1k: A large-scale dataset and benchmark for rgb-d object tracking. In: Pro-

ceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 3870–3878 (2023)