

PRVQL: Progressive Knowledge-guided Refinement for Robust Egocentric Visual Query Localization

Bing Fan¹ Yunhe Feng¹ Yapeng Tian² James Chenhao Liang³ Yuewei Lin⁴
Yan Huang¹ Heng Fan¹
¹University of North Texas ²University of Texas at Dallas
³U.S. Naval Research Laboratory ⁴Brookhaven National Laboratory

Abstract

Egocentric visual query localization (EgoVQL) focuses on localizing the target of interest in space and time from first-person videos, given a visual query. Despite recent progressive, existing methods often struggle to handle severe object appearance changes and cluttering background in the video due to lacking sufficient target cues, leading to degradation. Addressing this, we introduce **PRVQL**, a novel **Progressive knowledge-guided Refinement framework for EgoVQL**. The core is to continuously exploit target-relevant knowledge directly from videos and utilize it as guidance to refine both query and video features for improving target localization. Our PRVQL contains multiple processing stages. The target knowledge from one stage, comprising appearance and spatial knowledge extracted via two specially designed knowledge learning modules, are utilized as guidance to refine the query and videos features for the next stage, which are used to generate more accurate knowledge for further feature refinement. With such a progressive process, target knowledge in PRVQL can be gradually improved, which, in turn, leads to better refined query and video features for localization in the final stage. Compared to previous methods, our PRVQL, besides the given object cues, enjoys additional crucial target information from a video as guidance to refine features, and hence enhances EgoVQL in complicated scenes. In our experiments on challenging Ego4D, PRVQL achieves state-of-the-art result and largely surpasses other methods, showing its efficacy. Our code, model and results will be released at <https://github.com/fb-reps/PRVQL>.

1. Introduction

The egocentric visual query localization (EgoVQL) task [9] aims at answering the question “Where was the object X last seen in the video?”, with “ X ” being a visual query specified by a single image crop outside the search video. In specific, given a first-person video, its goal is to search and locate the

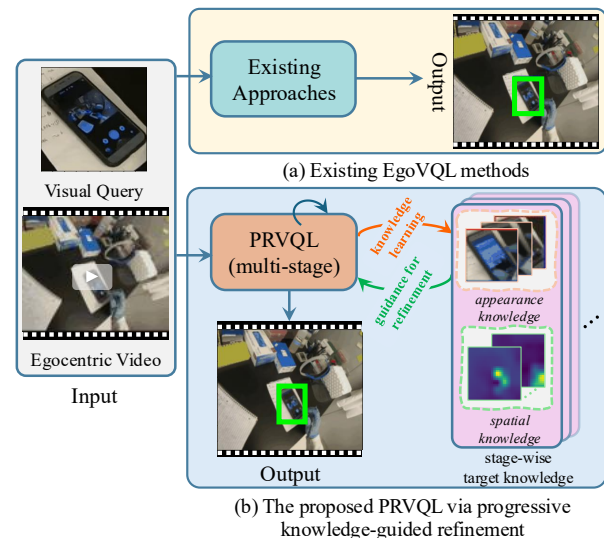


Figure 1. Comparison between current EgoVQL approaches in (a) and proposed PRVQL with progressive knowledge-guided refinement in (b). Best viewed in color and by zooming in for all figures.

visual query, *spatially* and *temporally*, by returning the most recent spatio-temporal tube. Owing to its important roles in numerous downstream object-centric applications including augmented and virtual reality, robotics, human-machine interaction, and so on, EgoVQL has drawn extensive attention from researchers in recent years.

Current approaches (e.g., [9, 14, 28, 29]) simply leverage the provided visual query as the *sole* cue to locate the target in the video (see Fig. 1 (a)). However, since the given visual query is cropped *outside* the search video, there often exists a *significant gap* between the query and the target of interest, due to rapid appearance variations in first-person videos caused by many factors, such as object pose change, motion blur, occlusion, and so forth. As a result, relying only on the given object query, as in existing methods, is *insufficient* to describe and distinguish the target from background in complicated scenarios with heavy appearance changes, resulting in performance degeneration. In addition, to achieve precise

localization, it is essential for an EgoVQL model to enhance target and meanwhile suppressing background regions from videos. Yet, this is often *overlooked* by existing approaches, making them easily suffer from cluttering background and therefore leading to suboptimal target localization.

The aforementioned issues faced by current methods naturally raise a question: *In addition to the given visual query, is there any other information that could be leveraged for enhancing EgoVQL?* We answer **yes**, and show the information directly explored from the *video itself*, as a supplement to the given target cue, is *effective* in improving EgoVQL.

Specifically, we propose a novel **Progressive knowledge-guided Refinement framework for EgoVQL (PRVQL)**. The core idea of our algorithm is to continuously exploit target-relevant knowledge from the video and leverage it to guide refinements of both query and video features, which are crucial for localization, for improving EgoVQL (see Fig. 1 (b)). Concretely, PRVQL consists of multiple processing stages. Each stage comprises two simple yet effective modules, including *appearance knowledge generation (AKG)* and *spatial knowledge generation (SKG)*. In specific, AKG works to mine visual information of the target from videos as the appearance knowledge. It first estimates potential target regions from a video using the query, and then selects top few highly confident regions to extract appearance knowledge from video features. Different from AKG, SKG focuses on exploring target position cues from videos as spatial knowledge by exploiting the readily available target-aware attention maps. In PRVQL, the appearance knowledge is used to guide the update of query feature, making it more discriminative, while the spatial knowledge is employed to enhance target and meanwhile suppressing unconcerned background in video features, enabling more focus on the target. The extracted appearance and spatial knowledge in one stage are used as guidance to respectively refine query and video features for next stage, which are adopted to learn more accurate knowledge for further feature refinement. Through this progressive process in PRVQL, the target knowledge can be gradually improved, which, in turn, results in better refined query and video features for target object localization in the final stage. Fig. 2 illustrates the framework of PRVQL.

To our best knowledge, PRVQL is the *first* method to exploit target-relevant appearance and spatial knowledge from the video to improve EgoVQL. Compared with existing solutions, PRVQL can leverage target information from both the given visual query and mined knowledge from the video for more robust localization. To verify its effectiveness, we conduct experiments on the challenging Ego4D [9], and our proposed PRVQL achieves state-of-the-art performance and largely outperforms other approaches, evidencing effectiveness of target knowledge for enhancing EgoVQL.

In summary, our main contributions are as follows: ♠ We propose a progressive knowledge-guided refinement frame-

work, dubbed PRVQL, that exploits knowledge from videos for improving EgoVQL; ♥ We introduce AKG for exploring visual information of target as appearance knowledge; ♣ We introduce SKG for learning spatial knowledge using target-aware attention maps; ♦ In extensive experiments on Ego4D, PRVQL achieves state-of-the-art performance.

2. Related Work

Egocentric Visual Query Localization. Egocentric visual query localization (EgoVQL) is an emerging and important computer vision task. Since its introduction in [9], EgoVQL has received extensive attention in recent years owing to its importance in numerous applications. Early methods [9, 28, 29] often utilize a bottom-up multi-stage framework, which sequentially and independently performs frame-level object detection, nearest peak temporal detection across the video, and bidirectional object tracking around the peak, to achieve EgoVQL. Despite the straightforwardness, this bottom-up design easily causes compounding errors across stages, thus degrading performance. Besides, the involvement of multiple detection and tracking components in this design leads to high complexities as well as inefficiency of the entire system, limiting its practicability. To deal with these issues, the recent method of [14] introduces a single-stage end-to-end framework for EgoVQL with Transformer [25], eliminating the need for multiple components and meanwhile showing promising performance. The method of [15] presents a training-free EgoVQL framework using foundational models and shows excellent results. Unlike existing models, the work of [20] introduces an online setting for EgoVQL.

In this work, we exploit target knowledge directly from the video to refine features for better localization. **Different** from previous works, PRVQL leverages cues from both the given query and mined target information for EgoVQL, significantly improving robustness, especially in presence of severe appearance variations and cluttering background.

Query-based Visual Localization. Query-based visual localization, broadly referring to localizing the target of interest from images or videos given a specific query (image or text), is a crucial problem in computer vision, and consists of a wide range of related tasks, including one-shot object detection [12, 32, 38], visual object tracking [1, 4, 17], visual grounding [6, 18, 39], spatio-temporal video grounding [10, 31], pedestrian search [16, 35], *etc.* Despite sharing some similarity with the above tasks in localizing the target, this work is **distinctive** by focusing on spatially and temporally searching for the target from egocentric videos, which is challenging due to frequent and heavy object appearance variations under the first-person views.

Progressive Learning Approach. Multi-stage progressive learning is a popular strategy to improve performance, and has been successfully applied for various tasks. For exam-

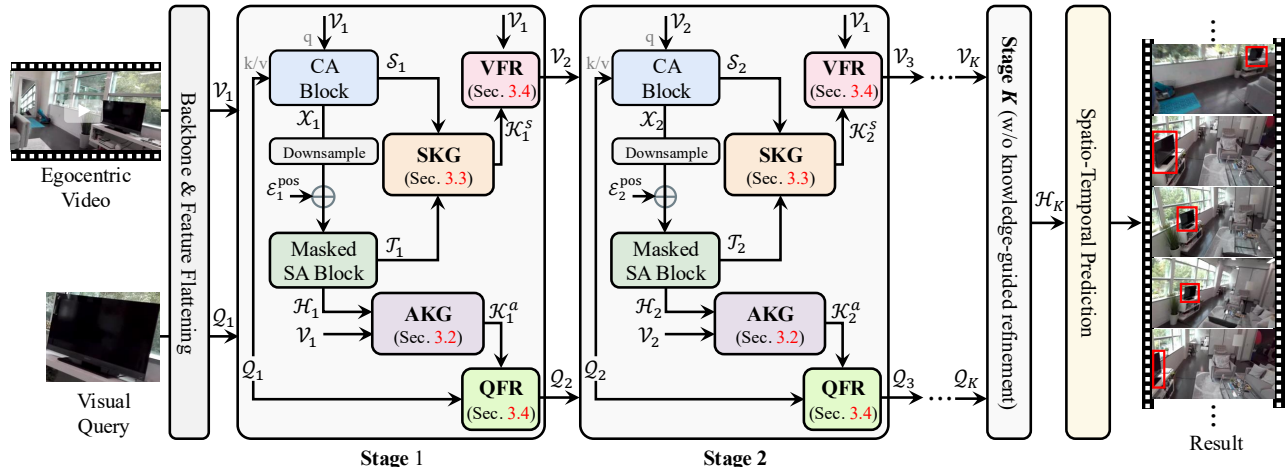


Figure 2. Overview of PRVQL, which aims to explore target knowledge directly from videos via AKG and SKG and applies it as guidance to refine query and video features with QFR and VFR for improving localization in EgoVQL through a multi-stage progressive architecture.

ple, the works of [2, 27, 34] introduce the cascade architecture to progressively refine the bounding boxes or features for improving object detection. The work in [33] presents a spatio-temporal progressive network for video action detection. The methods in [13, 37] introduce progressive refinement network for multi-scale semantic segmentation. The methods of [3, 36] apply progressive learning to improve features for saliency detection. The method in [8] proposes to progressively learn more accurate anchors for enhancing tracking. The work from [40] progressively transfers person pose for image generation. *Different* than these works, we focus on progressive refinement for improving EgoVQL.

3. The Proposed Method

Overview. In this paper, we propose PRVQL by exploiting crucial target knowledge directly from videos for improving target localization in EgoVQL. Our PRVQL is implemented as a progressive architecture. After feature extraction of the visual query and video frames, PRVQL performs iterative feature refinement guided by the target knowledge for localization through multiple stages (Sec. 3.1). As displayed in Fig. 2, each stage, expect for the final stage for prediction, consists of two crucial modules, comprising AKG (Sec. 3.2) and SKG (Sec. 3.3), for generating target appearance and spatial knowledge. The knowledge is leveraged as the guidance to refine query and video features (Sec. 3.4), which are applied in the next stage to generate more accurate target knowledge for further feature refinement. Through such a progressive process, the target knowledge can be gradually enhanced, which finally benefits learning more discriminative query and video features for improving EgoVQL.

3.1. Our PRVQL Framework

Visual Feature Extraction. In our PRVQL, we first extract features for the visual query and video frames. Specifically,

given the query q and a sequence of L frames $\mathcal{I} = \{I_i\}_{i=1}^L$ from the video, we utilize a shared backbone $\Phi(\cdot)$ [21] for extracting their features $\mathbf{q} = \Phi(q) \in \mathbb{R}^{H \times W \times C}$ and $F = \{\mathbf{f}_i\}_{i=1}^L$ with each $\mathbf{f}_i = \Phi(I(i)) \in \mathbb{R}^{H \times W \times C}$, where the H and W represent the spatial resolution of the features and C denotes the channel dimension. For subsequent processing, we flatten \mathbf{q} and F to obtain $\mathbf{Q} = \text{flatten}(\mathbf{q}) \in \mathbb{R}^{HW \times C}$ and $\mathbf{V} = \{\mathbf{v}_i\}_{i=1}^L$ with each $\mathbf{v}_i \in \mathbb{R}^{HW \times C}$.

Progressive Knowledge-guided Feature Refinement for EgoVQL. As mentioned earlier, the core idea of PRVQL is to exploit target knowledge directly from videos and apply it as guidance to enhance query and video features for target localization. For this purpose, PRVQL is implemented as a progressive architecture with multiple stages in a sequence. Each but the last stage involves target knowledge learning and knowledge-guided feature refinement, as in Fig. 2.

More specifically, for the k^{th} ($1 \leq k < K$) stage of our PRVQL, let \mathcal{Q}_k and \mathcal{V}_k denote the query and video features. For the first stage ($k = 1$), \mathcal{Q}_1 and \mathcal{V}_1 are initialized using query and video features extracted from the backbone, and $\mathcal{Q}_1 = \mathbf{Q}$ and $\mathcal{V}_1 = \mathbf{V}$. Otherwise, \mathcal{Q}_k and $\mathcal{V}_k = \{v_i^k\}_{i=1}^L$ are refined features in the last stage ($k - 1$). To mine target-specific knowledge from the video, we perform feature fusion between \mathcal{Q}_k and \mathcal{V}_k , aiming to inject target information into video feature for improving its target awareness. In specific, we leverage cross-attention from [25] for feature fusion owing to its powerfulness in feature modeling. Mathematically, this process can be expressed as follows,

$$\mathcal{X}_k = \{x_i^k | x_i^k = \text{CAB}(v_i^k, \mathcal{Q}_k)\} \quad i = 1, 2, \dots, L \quad (1)$$

where \mathcal{X}_k is the fused feature in stage k , and v_i^k the feature in frame i . $\text{CAB}(\mathbf{z}, \mathbf{u})$ is the cross-attention (CA) block with \mathbf{z} generating query and \mathbf{u} key/value. Due to space limitation, please see *supplementary material* for detailed architecture. Besides fused feature, we also obtain target-aware

spatial attention maps $\mathcal{S}_k = \{s_i^k\}_{i=1}^L \in \mathbb{R}^{L \times HW \times HW}$ for L frames in Eq. (1), with each $s_i^k \in \mathbb{R}^{HW \times HW}$ the attention map from the cross-attention operation in $\text{CAB}(v_i^k, \mathcal{Q}_k)$.

To further capture spatio-temporal relations from videos for enhancing features, we apply self-attention [25] on \mathcal{X}_k by propagating the query information spatially and temporally. Considering that targets in nearby frames are highly correlated, we restrict the attention operation in a temporal window using a masking strategy, similar to [14]. To reduce the computation, we downsample \mathcal{X}_k to decrease the spatial dimension of each frame feature to $h \times w$. Then, we add a position embedding $\mathcal{E}_k^{\text{pos}}$ to the video feature. This process can be written as follows,

$$\tilde{\mathcal{X}}_k = \text{Downsample}(\mathcal{X}_k) + \mathcal{E}_k^{\text{pos}} \quad (2)$$

where $\text{Downsample}(\cdot)$ represents the downsampling operation implemented with convolution operation. Afterwards, masked self-attention is applied on as $\tilde{\mathcal{X}}$ as follows,

$$\mathcal{H}_k = \text{MaskedSA}(\tilde{\mathcal{X}}_k) \quad (3)$$

where \mathcal{H}_k represents enhanced video feature. $\text{MaskedSA}(\mathbf{z})$ denotes the masked self-attention block with \mathbf{z} generating query/key/value. In this block, each feature element from frame i only attends to feature elements from frames in the temporal range $[(i - u), (i + u)]$, which can be easily implemented using masking strategy [5, 25]. From Eq. (3), besides the \mathcal{H}_k , we also gain the temporal-aware spatial attention maps, denoted as $\mathcal{T}_k \in \mathbb{R}^{L \times h \times w \times L \times h \times w}$, for the target in the video, which will be used for knowledge generation.

With video feature \mathcal{H}_k and attention maps \mathcal{S}_k and \mathcal{T}_k , the target knowledge can be extracted with the AKG and SKG modules (as explained later in Sec. 3.2 and 3.3), as follows,

$$\mathcal{K}_k^a = \text{AKG}(\mathcal{H}_k, \mathcal{V}_k) \quad \mathcal{K}_k^s = \text{SKG}(\mathcal{S}_k, \mathcal{T}_k) \quad (4)$$

where \mathcal{K}_k^a represents the appearance knowledge and \mathcal{K}_k^s the spatial knowledge. Guided by \mathcal{K}_k^a and \mathcal{K}_k^s in stage k , we can refine query and video features using two update modules QFR and VFR (as described later in Sec. 3.4) as follows,

$$\mathcal{Q}_{k+1} = \text{QFR}(\mathcal{K}_k^a, \mathcal{Q}_k) \quad \mathcal{V}_{k+1} = \text{VFR}(\mathcal{K}_k^s, \mathcal{V}_1) \quad (5)$$

where \mathcal{Q}_{k+1} and \mathcal{V}_{k+1} are refined features guided by target knowledge, which are fed to the next stage ($k + 1$) to generate more accurate knowledge for further feature refinement. Fig. 3 compares the attention maps from the masked self-attention with and without using our approach. We can see that, our method with refined features guided by knowledge can better focus on the target in the video and thus improves target localization, showing its efficacy.

For the final K^{th} stage in PRVQL, since no knowledge is extracted, the AKG and SKG modules are removed. Given the visual query and video features \mathcal{Q}_K and \mathcal{V}_K from the $(K - 1)^{\text{th}}$ stage, we can then obtain the final enhanced video feature \mathcal{H}_K through Eqs. (1)-(3) in the K^{th} stage. With \mathcal{H}_K ,

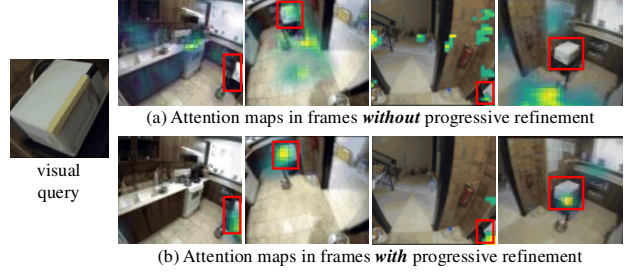


Figure 3. Comparison of attention maps for video frames from the masked self-attention block *without* (a) and *with* (b) our progressive refinement. As shown, our method can better focus on the target regions, and hence can improve target localization in EgoVQL. The red boxes indicate the foreground object to localize.

we use the prediction heads as in [14] for target localization via regression and classification. For details of the adopted prediction heads, please kindly refer to [14].

3.2. Appearance Knowledge Generation (AKG)

In order to extract discriminative visual information of target directly from the video, we introduce a simple yet highly effective module, named *appearance knowledge generation* (AKG), for appearance knowledge learning. Specifically, it first estimates the potential target regions from the video using target-aware video features. Then, based on confidence scores of these regions, we select the top few ones to extract target features from the video as the appearance knowledge.

Specifically, given the target-aware video feature \mathcal{H}_k , we first reshape it to the 2D feature map, and then increase its spatial resolution back to $H \times W$ as follows,

$$\tilde{\mathcal{H}}_k = \text{Upsample}(\text{Reshape}(\mathcal{H}_k)) \quad (6)$$

where $\text{Upsample}(\cdot)$ denotes the upsampling operation. After this, we apply $\tilde{\mathcal{H}}_k$ to produce temporal confidence scores and spatial box regions for target in each frame. More concretely, we first split $\tilde{\mathcal{H}}_k$ along the channel dimension into two equal halves $\tilde{\mathcal{H}}_k^t$ and $\tilde{\mathcal{H}}_k^s$ via $\tilde{\mathcal{H}}_k^t, \tilde{\mathcal{H}}_k^s = \text{Split}(\tilde{\mathcal{H}}_k)$. Inspired by [14], we perform classification and regression to predict temporal confidence scores and spatial boxes using multi-scale anchors [23]. Specifically, two Conv blocks are applied on $\tilde{\mathcal{H}}_k^t(i)$ and $\tilde{\mathcal{H}}_k^s(i)$ for prediction as follows,

$$\tilde{\mathcal{C}}_k = \text{ConvBlock}(\tilde{\mathcal{H}}_k^t) \quad \Delta\tilde{\mathcal{B}}_k = \text{ConvBlock}(\tilde{\mathcal{H}}_k^s) \quad (7)$$

where $\tilde{\mathcal{C}}_k \in \mathbb{R}^{L \times H \times W \times m}$ denotes the temporal confidence scores for target in L frames with m the number of anchors at each position. $\Delta\tilde{\mathcal{B}}_k \in \mathbb{R}^{L \times H \times W \times 4m}$ is the offsets to the anchor boxes $\tilde{\mathcal{B}}$, and target boxes $\tilde{\mathcal{B}}_k = \Delta\tilde{\mathcal{B}}_k + \tilde{\mathcal{B}}$. With $\tilde{\mathcal{C}}_k$, the confidence score in each frame is determined by the highest classification score of anchors, and the target region is the box corresponding to the box with the highest classification score. This way, we can obtain the confidence scores

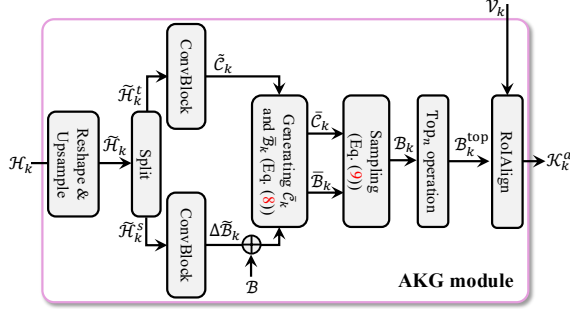


Figure 4. Illustration of appearance knowledge generation (AKG).

\bar{C}_k and target regions \bar{B}_k in each frames as follows,

$$\begin{aligned}\bar{C}_k &= \{c_k^i | c_k^i, d_k^i = \text{Max}(\tilde{C}_k(i))\} \\ \bar{B}_k &= \{b_k^i | b_k^i = \text{Index}(\tilde{B}_k(i), d_k^i)\}\end{aligned}\quad (8)$$

where $i \in [1, L]$ is the frame index. c_k^i is the highest value selected from the classification scores $\tilde{C}_k(i) \in \mathbb{R}^{H \times W \times m}$ of anchors in frame i , and d_k^i is its index. b_k^i is the target box corresponding to c_k^i in frame, and extracted from $\tilde{B}_k(i) \in \mathbb{R}^{H \times W \times 4m}$. $\text{Max}(\cdot)$ is to select the maximum and its index, and $\text{Index}(\cdot)$ to extract the box from $\tilde{B}_k(i)$ given its index.

With $\bar{C}_k \in \mathbb{R}^{L \times 1}$ and $\bar{B}_k \in \mathbb{R}^{L \times 4}$, we first sample target regions with high confidence scores as follows,

$$\mathcal{B}_k = \text{Sample}(\bar{B}_k(i), \bar{C}_k(i), \tau) = \{\bar{B}_k(i) | \bar{C}_k(i) > \tau\} \quad (9)$$

Then, we extract n regions from \mathcal{B}_k with the top confidence scores via $\mathcal{B}_k^{\text{top}} = \text{Top}_n(\mathcal{B}_k)$. If the number of regions in \mathcal{B}_k is less than n , we keep all regions. After this, RoIAlign [11] is used to extract the appearance knowledge from \mathcal{V}_k via

$$\mathcal{K}_k^a = \text{RoIAlign}(\mathcal{V}_k, \mathcal{B}_k^{\text{top}}) \quad (10)$$

where \mathcal{K}_k^a represents the appearance knowledge from AKG in the k^{th} stage. Please notice that, in Eq. (10), we only perform RoIAlign in frames corresponding to $\mathcal{B}_k^{\text{top}}$. Since \mathcal{K}_k^a is generated from the video itself, when using it as guidance to refine the query feature, we can reduce the discrepancy between the query and the foreground target. By deploying AKG in each but the last stage, \mathcal{K}_k^a could be gradually improved with better refined query feature in each stage. Fig. 4 illustrates AKG for appearance knowledge generation.

3.3. Spatial Knowledge Generation (SKG)

In addition to appearance knowledge, we explore target spatial knowledge from the video for improving video features. Specifically, inspired by the *observation* that intermediate attention maps from previous attention operations reflect the spatial cues of target in each frame to some extent, similar to the concept of “*saliency*” but for the target, we propose the *spatial knowledge generation* (SKG) module, which works to leverage readily available attention maps as guidance for enhancing target while suppressing background in the video

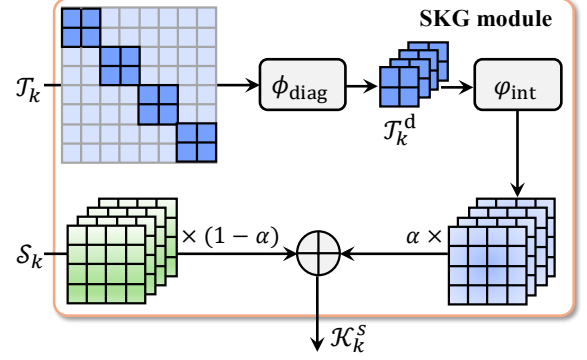


Figure 5. Illustration of spatial knowledge generation (SKG).

features, enabling more focus on the target in PRVQL.

Concretely in our SKG, we exploit the target-aware spatial attention maps \mathcal{S}_k from cross-attention block in Eq. (1) and temporal-aware spatial attention maps \mathcal{T}_k from masked self-attention block in Eq. (3) for spatial knowledge learning. Specifically, given \mathcal{S}_k and \mathcal{T}_k , we first extract the inter-frame spatial attention maps \mathcal{T}_k^d by extracting diagonal elements from \mathcal{T}_k as follows,

$$\mathcal{T}_k^d = \phi_{\text{diag}}(\mathcal{T}_k) = \{t_i^k\}_{i=1}^L \quad (11)$$

where ϕ_{diag} denotes the operation to extract diagonal elements, and $t_i^k \in \mathbb{R}^{hw \times hw}$ represents the attention maps for frame i . To match the spatial dimension of \mathcal{T}_k^d and \mathcal{S}_k , we first perform bilinear interpolation on \mathcal{T}_k^d to increase its spatial resolution to $HW \times HW$, and then combines these two attention maps to obtain spatial knowledge. Mathematically, this process can be expressed as follows,

$$\mathcal{K}_k^s = \alpha \cdot \varphi_{\text{int}}(\mathcal{T}_k^d) + (1 - \alpha) \cdot \mathcal{S}_k \quad (12)$$

where φ_{int} denotes the bilinear interpolation operation, \mathcal{K}_k^s is the target spatial knowledge, and α is a parameter to balance different attention maps. Since \mathcal{K}_k^s indicates the target position cues in each frame in some degree, we can use it to highlight target while restraining background in videos for improving localization. Similar to AKG, SKG is deployed in each but the last stage of PRVQL. Fig. 5 illustrates SKG.

3.4. Feature Refinement with Knowledge

With target appearance knowledge \mathcal{K}_k^a and spatial knowledge \mathcal{K}_k^s obtained from AKG and SKG in stage k ($1 < k \leq K$), we then apply them as guidance to refine the query and video features through *query feature refinement* (QFR) and *video feature refinement* (VFR) modules.

Query Feature Refinement (QFR). QFR aims to refine the query feature with guidance from learned target appearance knowledge. Specifically, it adopts a cross-attention block to fuse appearance knowledge \mathcal{K}_k^a into the query. Given the query feature \mathcal{Q}_k and appearance knowledge \mathcal{K}_k^a in stage k , we first apply a Conv block on \mathcal{K}_k^a and then perform refine-

ment via QFR as follows,

$$\mathcal{Q}_{k+1} = \text{QFR}(\mathcal{Q}_k, \mathcal{K}_k^a) = \text{CAB}(\mathcal{Q}_k, \text{CNB}(\mathcal{K}_k^a)) \quad (13)$$

where \mathcal{Q}_{k+1} is refined query feature and fed to next stage for learning more accurate knowledge, which in turn leads to better query feature for localization in the final stage. It is worth noting that, besides cross-attention, we explore different strategies to combine \mathcal{Q}_k and \mathcal{K}_k^a , including addition and concatenation operations. We observe that using cross-attention achieves the best performance, as exhibited in our experiments provided in the *supplementary material*.

Video Feature Refinement (VFR). VFR uses the target spatial knowledge to refine initial video feature by enhancing target while suppressing the background. Given the initial video feature \mathcal{V}_1 and learn spatial knowledge \mathcal{K}_k^s in stage k , we use residual connection to refine \mathcal{V}_1 as follows,

$$\mathcal{V}_{k+1} = \beta \cdot (\mathcal{K}_k^s \odot \mathcal{V}_1) + (1 - \beta) \cdot \mathcal{V}_1 \quad (14)$$

where \mathcal{V}_{k+1} denotes the refined video feature that is used for the next stage, β is a balancing parameter, and \odot represents the pixel-wise multiplication.

3.5. Optimization and Inference

Optimization. Given a video and a visual query, we predict confidence scores $\tilde{\mathcal{C}}_k$ and target boxes $\tilde{\mathcal{B}}_k$ ($\tilde{\mathcal{B}}_k = \Delta\tilde{\mathcal{B}}_k + \tilde{\mathcal{B}}$) in each stage k ($1 \leq k \leq K$). During training, given the groundtruth boxes \mathcal{B}^* and temporal occurrence scores \mathcal{S}^* , we design the following loss function \mathcal{L}_k for stage k ,

$$\mathcal{L}_k = \mathcal{L}_{L_1}(\tilde{\mathcal{B}}_k, \mathcal{B}^*) + \lambda_1 \mathcal{L}_{\text{GIoU}}(\tilde{\mathcal{B}}_k, \mathcal{B}^*) + \lambda_2 \mathcal{L}_{\text{BCE}}(\tilde{\mathcal{S}}_k, \mathcal{S}^*) \quad (15)$$

where \mathcal{L}_{L_1} , $\mathcal{L}_{\text{GIoU}}$, and \mathcal{L}_{BCE} represent the L_1 loss, generalized IoU (GIoU) [24] loss, and binary cross-entropy (BCE) loss, respectively. λ_1 and λ_2 are two balancing parameters. With Eq. (15), the total training loss $\mathcal{L}_{\text{total}}$ can be obtained via $\mathcal{L}_{\text{total}} = \sum_{k=1}^K \mathcal{L}_k$. Following [9, 14, 29], we perform hard negative mining during training to decrease false positive prediction. For details, please refer to [9, 14, 29].

Inference. We employ the same strategy as in [14] to obtain the prediction result. Specifically, during inference, we first obtain the target region in each frame by selecting target box with the highest confidence score. Please note that, the target regions with confidences scores smaller than a threshold, set to 0.79, will be discarded. After this, we select the most recent peak and generate a response track via bidirectional search from the peak. Details can be seen in [14].

4. Experiments

Implementation. Our PRVQL is implemented using PyTorch [22]. Similar to [14], we apply the ViT [7] pretrained with DINOv2 [21] as the backbone. PRVQL is end-to-end trained for 50 epoches (60K iterations in total) with a batch size of 12, utilizing the AdamW optimizer [19] with a peak

Table 1. Comparison on the Ego4D validation set.

Methods	tAP ₂₅	stAP ₂₅	rec%	Succ
STARK [ICCV'21]	0.10	0.04	12.41	18.70
SiamRCNN [CVPR'22]	0.22	0.15	32.92	43.24
NFM [VQ2D Challenge'22]	0.26	0.19	37.88	47.90
CocoFormer [CVPR'23]	0.26	0.19	37.67	47.68
VQLoC [NeurIPS'23]	0.31	0.22	47.05	55.89
PRVQL (ours)	0.35	0.27	47.87	57.93

Table 2. Comparison on the Ego4D test set.

Methods	tAP ₂₅	stAP ₂₅	rec%	Succ
SiamRCNN [CVPR'22]	0.21	0.13	34.0	41.60
NFM [VQ2D Challenge'22]	0.24	0.17	36.38	45.07
CocoFormer [CVPR'23]	0.26	0.18	43.20	48.10
VQLoC [NeurIPS'23]	0.32	0.24	45.11	55.88
PRVQL (ours)	0.37	0.28	45.70	59.43

Table 3. Comparison of speed on Ego4D.

	STARK	SiamRCNN	NFM	CocoFormer	VQLoC	PRVQL
FPS	33	3	3	3	36	30

learning rate of 10^{-4} and a weight decay of 5×10^{-2} . The query image and video frames are resized to 480×480 . The number of stages K in PRVQL is empirically set to 3, and the pooling size for RoIAlign is 5. The number of selected boxes n for appearance knowledge is 3, and the threshold τ is set to 0.7. The parameters α and β are empirically set to 0.5 and 0.1. λ_1 and λ_2 are 0.3 and 100. The video frame length L , similar to [14], is set to 30 with frames randomly selected to ensure coverage of at least a portion of the response track. For the anchor boxes in localization, we employ four scales (16^2 , 32^2 , 64^2 , 48^2) with three different aspect ratios (0.5, 1, 2) for each anchor box, similar to [14].

4.1. Dataset and Evaluation Metrics

Dataset. Following [14, 29], we conduct the experiments on Ego4D [9], a recently proposed large-scale dataset dedicated to first-person video understanding. Similar to [14], we use videos from the VQ2D task, with 13.6K, 4.5K, 4.4K pairs of queries and videos for training, validation, and test.

Evaluation Metrics. Following [14, 29], we adopt the metrics provided by Ego4D [9] for evaluation, including temporal average precision (tAP₂₅), spatio-temporal average precision (stAP₂₅), recovery (rec%), and success (Succ). For more details of these metrics, please refer to [9].

4.2. State-of-the-art Comparison

To verify the effectiveness of PRVQL, we compare it with other methods on Ego4D, including STARK [30], SiamRCNN [26], NFM [28], CocoFormer [29], and VQLoC [14]. Tab. 1 displays results on the Ego4D validate test. As in Tab. 1, we can clearly see that PRVQL achieves the best performance on all four metrics. In particular, it achieves 0.35

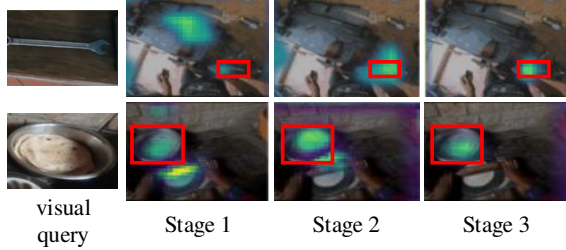


Figure 6. Visualization of refinement in different stages. The red bounding boxes indicate the groundtruth.

Table 4. Ablation studies of AKG and SKG.

	AKG	SKG	tAP ₂₅	stAP ₂₅	rec%	Succ
❶	-	-	0.32	0.23	45.24	55.37
❷	✓	-	0.34	0.26	47.34	57.27
❸	-	✓	0.33	0.24	46.33	56.46
❹	✓	✓	0.35	0.27	47.87	57.93

Table 5. Ablation studies on the number of stages.

	# Stages	tAP ₂₅	stAP ₂₅	rec%	Succ
❶	$K = 1$	0.32	0.23	45.24	55.37
❷	$K = 2$	0.34	0.27	47.25	56.43
❸	$K = 3$	0.35	0.27	47.87	57.93
❹	$K = 4$	0.33	0.26	45.91	55.29

tAP₂₅ and 0.27 stAP₂₅ scores, which outperforms the second best VQLoC with 0.31 tAP₂₅ and 0.22 stAP₂₅ scores by 4% and 5%. Besides, the rec and Succ scores of PRVQL are 47.87% and 57.93 respectively, which surpasses the 47.05% rec and 55.89 Succ scores of VQLoC, evidencing its effectiveness. Besides, in Tab. 2, we further report the experimental results on Ego4D test set. As in Tab. 2, PRVQL again achieves the best results on all four metrics. Specifically, PRVQL obtains 0.37 tAP₂₅ and 0.28 stAP₂₅ scores. Compared to VQLoC, our approach shows gains of 5% and 4%, respectively, on tAP₂₅ and stAP₂₅. All these show the efficacy of target knowledge in improving EgoVQL.

Moreover, we show the speed, measured by frames per second (FPS), for different methods in Tab. 3. PRVQL runs fast at 30 FPS. Despite being slightly slower than VQLoC running at 36 FPS, PRVQL is more robust in localization, showing a better balance between accuracy and speed.

4.3. Analysis of Refinement by Multiple Stages

Our PRVQL improves the localization performance through multiple stages of refinement guided with target knowledge. In Fig. 6, we show the attention maps of the visual query from different stages of PRVQL in the video frame. From Fig. 6, we can clearly see that, the attention maps are gradually focused on the target regions via refinement in multiple stages and benefit better target localization, which evidences the effectiveness of our PRVQL.

Table 6. Ablation studies on the threshold τ .

	Threshold	tAP ₂₅	stAP ₂₅	rec%	Succ
❶	$\tau = 0.6$	0.32	0.24	44.82	55.97
❷	$\tau = 0.7$	0.35	0.27	47.87	57.93
❸	$\tau = 0.8$	0.34	0.26	46.53	57.33

Table 7. Ablation studies on the number of target boxes in AKG.

		tAP ₂₅	stAP ₂₅	rec%	Succ
❶	$n = 2$	0.34	0.26	47.27	56.03
❷	$n = 3$	0.35	0.27	47.87	57.93
❸	$n = 4$	0.36	0.26	46.59	57.62
❹	$n = 5$	0.35	0.24	46.84	56.95

Table 8. Ablation studies on RoIAlign feature size.

	Size	tAP ₂₅	stAP ₂₅	rec%	Succ
❶	3	0.33	0.26	46.94	56.06
❷	5	0.35	0.27	47.87	57.93
❸	7	0.34	0.27	47.58	57.38
❹	9	0.32	0.25	46.37	55.27

4.4. Ablation Study

For better understanding of PRVQL, we conduct extensive ablation studies on Ego4D validation set as follows.

Impact of AKG and SKG. AKG and SKG are two key modules in PRVQL for knowledge generation. To analyze these two modules, we conduct thorough ablation studies in Tab. 4. From Tab. 4, we can see that, without AKG and SKG, the tAP₂₅ and stAP₂₅ scores are 0.32 and 0.23, respectively (❶). By applying AKG alone for refinement with appearance knowledge, they are significantly improved to 0.34 and 0.26 with performance gains of 0.02 and 0.03 (❷ v.s. ❶). When using only SKG for refinement with spatial knowledge, tAP₂₅ and stAP₂₅ are improved to 0.33 and 0.24 (❸ v.s. ❶). From this table, we also observe that, using appearance knowledge for refinement in PRVQL brings more gains than the spatial knowledge (❷ v.s. ❸). When using both AKG and SKG in our PRVQL, we achieve the best performance with 0.35 tAP₂₅ and 0.27 stAP₂₅ scores (❹ v.s. ❶), which clearly evidences the efficacy of target knowledge for improving the robustness of EgoVQL.

Impact of the number of stages. PRVQL is a progressive architecture with K stages to explore the target knowledge for refinement. We conduct an ablation study on K in PRVQL in Tab. 5. From Tab. 5, we observe that, when setting $K = 1$, which means only one stage is used and the target knowledge is not used due to one-stage design, the tAP₂₅ and stAP₂₅ scores are 0.32 and 0.23 (❶). When adding the second stage, tAP₂₅ and stAP₂₅ are largely improved by 2% and 4% to 0.34 and 0.27, respectively (❷). With three stages, the tAP₂₅ and stAP₂₅ scores are further boosted to 0.35 and 0.27 (❸). When setting $K = 4$ with 4 stages, the performance is decreased with 0.33 tAP₂₅ and

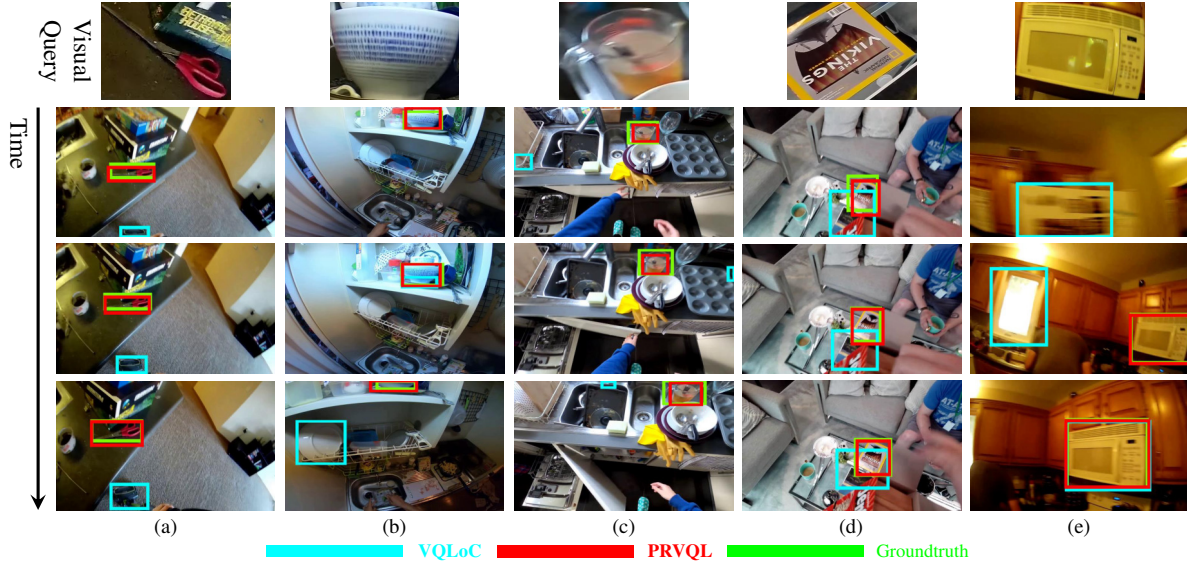


Figure 7. Qualitative analysis and comparison between our PRVQL and state-of-the-art VQLoC in representative videos with different challenges. We observe that, owing to our target knowledge from videos, PRVQL can more robustly localize the target of interest.

0.26 stAP_{25} scores (Ⓔ). Hence, we set K to 3 in PRVQL.

Impact of threshold τ in AKG. The threshold τ is used to filter out less confident target regions in AKG, aiming to avoid noisy features in appearance knowledge generation. In this work, we conduct an ablation to study the impact of τ on the final performance in Tab. 6. As shown in Tab. 6, we can see that, when setting τ to 0.7, PRVQL achieves the best performance on all four metrics (Ⓔ).

Impact of number of target boxes in AKG. In AKG, we extract visual features from the top n highly confident target regions for appearance knowledge generation. We conduct an ablation on n in Tab. 7. From Tab. 7, we can observe that, when using the top 3 target regions for knowledge learning in AKG, we achieve the best overall performance (Ⓔ).

Impact of RoIAlign Feature Size. With the top n selected target regions, we perform the RoIAlign operation [11] to obtain target appearance knowledge. The RoIAlign feature size may have an impact on the target appearance knowledge. A too small size may result in the coarse spatial information of the target, while a too large size may lead to losing discriminative local features for the target, both degrading performance. In this work, we study different RoIAlign feature sizes in Tab. 8. As shown, when setting the size to 5 in RoIAlign, PRVQL shows the best overall performance.

4.5. Qualitative Analysis

To provide further analysis of our PRVQL, we show the visualization results of its localization and compare it with VQLoC in Fig. 7. Specifically, we show results on several representative videos, including video in (a) with *pose variation*, video in (b) with *cluttering background* and *out-of-view*, video in (c) with *occlusion* and *low resolution*, video

in (d) with *pose variation* and *cluttering background*, and video in (e) with *motion blur* and *distractor*. From Fig. 7, we can see that, our method can robustly and accurately localize the target in all these challenges, owing to the help of target knowledge from videos, while VQLoC is prone to drift to the background due to lack of discriminative target information, evidencing the effectiveness of our method.

Due to limited space, we demonstrate more results, analysis, and ablation studies in the *supplementary material*.

5. Conclusion

In this paper, we present a multi-stage architecture, dubbed PRVQL, for improving EgoVQL via exploring the target knowledge to refine features for robust localization. In each stage, two key modules, including AKG and SKG, are used to extract target appearance and spatial knowledge from the video. The knowledge from one stage is used as guidance to refine query and video features in the next stage, which are adopted for learning more accurate knowledge for further refinement. Through this progressive process, PRVQL learns gradually improved knowledge, which in turn leads to better refined features for localization in the final stage. Our experiments on Ego4D show that PRVQL achieves promising results and largely surpasses other methods.

Acknowledgment. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of U.S. Naval Research Laboratory (NRL) or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

- [1] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *ECCVW*, 2016. 2
- [2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018. 3
- [3] Shuhan Chen and Yun Fu. Progressively guided alternate refinement network for rgb-d salient object detection. In *ECCV*, 2020. 3
- [4] Xin Chen, Houwen Peng, Dong Wang, Huchuan Lu, and Han Hu. Seqtrack: Sequence to sequence learning for visual object tracking. In *CVPR*, 2023. 2
- [5] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 4
- [6] Jiajun Deng, Zhengyuan Yang, Tianlang Chen, Wengang Zhou, and Houqiang Li. Transvg: End-to-end visual grounding with transformers. In *ICCV*, 2021. 2
- [7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 6
- [8] Heng Fan and Haibin Ling. Siamese cascaded region proposal networks for real-time visual tracking. In *CVPR*, 2019. 3
- [9] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022. 1, 2, 6
- [10] Xin Gu, Heng Fan, Yan Huang, Tiejian Luo, and Libo Zhang. Context-guided spatio-temporal video grounding. In *CVPR*, 2024. 2
- [11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 5, 8
- [12] Ting-I Hsieh, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. One-shot object detection with co-attention and co-excitation. In *NeurIPS*, 2019. 2
- [13] Chuong Huynh, Anh Tuan Tran, Khoa Luu, and Minh Hoai. Progressive semantic segmentation. In *CVPR*, 2021. 3
- [14] Hanwen Jiang, Santhosh Kumar Ramakrishnan, and Kristen Grauman. Single-stage visual query localization in egocentric videos. *NeurIPS*, 2023. 1, 2, 4, 6
- [15] Savya Khosla, Alexander Schwing, Derek Hoiem, et al. Relocate: A simple training-free baseline for visual query localization using region-based representations. In *CVPR*, 2025. 2
- [16] Shuang Li, Tong Xiao, Hongsheng Li, Bolei Zhou, Dayu Yue, and Xiaogang Wang. Person search with natural language description. In *CVPR*, 2017. 2
- [17] Liting Lin, Heng Fan, Zhipeng Zhang, Yaowei Wang, Yong Xu, and Haibin Ling. Tracking meets lora: Faster training, larger model, stronger performance. In *ECCV*, 2024. 2
- [18] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *ECCV*, 2024. 2
- [19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 6
- [20] Zaira Manigrasso, Matteo Dunnhofer, Antonino Furnari, Moritz Nottebaum, Antonio Finocchiaro, Davide Marana, Giovanni Maria Farinella, and Christian Micheloni. Online episodic memory visual query localization with egocentric streaming object memory. *arXiv*, 2024. 2
- [21] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. 3, 6
- [22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 6
- [23] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NIPS*, 2015. 4
- [24] 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 *CVPR*, 2019. 6
- [25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 2, 3, 4
- [26] Paul Voigtlaender, Jonathon Luiten, Philip HS Torr, and Bastian Leibe. Siam r-cnn: Visual tracking by re-detection. In *CVPR*, 2020. 6
- [27] Thang Vu, Hyunjun Jang, Trung X Pham, and Chang Yoo. Cascade rpn: Delving into high-quality region proposal network with adaptive convolution. In *NeurIPS*, 2019. 3
- [28] Mengmeng Xu, Cheng-Yang Fu, Yanghao Li, Bernard Ghanem, Juan-Manuel Perez-Rua, and Tao Xiang. Negative frames matter in egocentric visual query 2d localization. *arXiv*, 2022. 1, 2, 6
- [29] Mengmeng Xu, Yanghao Li, Cheng-Yang Fu, Bernard Ghanem, Tao Xiang, and Juan-Manuel Pérez-Rúa. Where is my wallet? modeling object proposal sets for egocentric visual query localization. In *CVPR*, 2023. 1, 2, 6
- [30] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. In *ICCV*, 2021. 6
- [31] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Tubedet: Spatio-temporal video grounding with transformers. In *CVPR*, 2022. 2

- [32] Hanqing Yang, Sijia Cai, Hualian Sheng, Bing Deng, Jianqiang Huang, Xian-Sheng Hua, Yong Tang, and Yu Zhang. Balanced and hierarchical relation learning for one-shot object detection. In *CVPR*, 2022. 2
- [33] Xitong Yang, Xiaodong Yang, Ming-Yu Liu, Fanyi Xiao, Larry S Davis, and Jan Kautz. Step: Spatio-temporal progressive learning for video action detection. In *CVPR*, 2019. 3
- [34] Mingqiao Ye, Lei Ke, Siyuan Li, Yu-Wing Tai, Chi-Keung Tang, Martin Danelljan, and Fisher Yu. Cascade-detr: delving into high-quality universal object detection. In *ICCV*, 2023. 3
- [35] Rui Yu, Dawei Du, Rodney LaLonde, Daniel Davila, Christopher Funk, Anthony Hoogs, and Brian Clipp. Cascade transformers for end-to-end person search. In *CVPR*, 2022. 2
- [36] Xiaoning Zhang, Tiantian Wang, Jinqing Qi, Huchuan Lu, and Gang Wang. Progressive attention guided recurrent network for salient object detection. In *CVPR*, 2018. 3
- [37] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnets for real-time semantic segmentation on high-resolution images. In *ECCV*, 2018. 3
- [38] Yizhou Zhao, Xun Guo, and Yan Lu. Semantic-aligned fusion transformer for one-shot object detection. In *CVPR*, 2022. 2
- [39] Chaoyang Zhu, Yiyi Zhou, Yunhang Shen, Gen Luo, Xingjia Pan, Mingbao Lin, Chao Chen, Liujuan Cao, Xiaoshuai Sun, and Rongrong Ji. Seqtr: A simple yet universal network for visual grounding. In *ECCV*, 2022. 2
- [40] Zhen Zhu, Tengpeng Huang, Baoguang Shi, Miao Yu, Bofei Wang, and Xiang Bai. Progressive pose attention transfer for person image generation. In *CVPR*, 2019. 3