VISAGE: Video Instance Segmentation with Appearance-Guided Enhancement

Hanjung Kim[1], Jaehyun Kang[1], Miran Heo[1], Sukjun Hwang[2], Seoung Wug Oh[3], and Seon Joo Kim[1]

[1] Yonsei University
[2] Carnegie Mellon University
[3] Adobe Research

Abstract. In recent years, online Video Instance Segmentation (VIS) methods have shown remarkable advancement with their powerful query-based detectors. Utilizing the output queries of the detector at the frame-level, these methods achieve high accuracy on challenging benchmarks. However, our observations demonstrate that these methods heavily rely on location information, which often causes incorrect associations between objects. This paper presents that a key axis of object matching in trackers is appearance information, which becomes greatly instructive under conditions where positional cues are insufficient for distinguishing their identities. Therefore, we suggest a simple yet powerful extension to object decoders that explicitly extract embeddings from backbone features and drive queries to capture the appearances of objects, which greatly enhances instance association accuracy. Furthermore, recognizing the limitations of existing benchmarks in fully evaluating appearance awareness, we have constructed a synthetic dataset to rigorously validate our method. By effectively resolving the over-reliance on location information, we achieve state-of-the-art results on YouTube-VIS 2019/2021 and Occluded VIS (OVIS). Code is available at https://github.com/KimHanjung/VISAGE.

Keywords: video instance segmentation, object appearance

1 Introduction

Video Instance Segmentation (VIS) is a challenging task that requires classification, segmentation, and tracking of distinct instances throughout a video sequence [32]. Current studies in VIS can be primarily categorized into two approaches: online and offline, based on whether a video is processed in a per-frame or per-clip manner. Recently, the advancement of frame-level object detectors has resulted in online methods becoming increasingly dominant in the VIS field.

As detectors directly impact the accuracy in the video domain, recent online models are primarily built using the powerful query-based detectors [5, 39]. Spatially decoding image information, the detectors are designed to represent object-wise information using the queries. Therefore, the online VIS methods reuse these queries from the detectors and have achieved substantial improvements in multiple challenging benchmarks [26, 34] by mostly adopting either

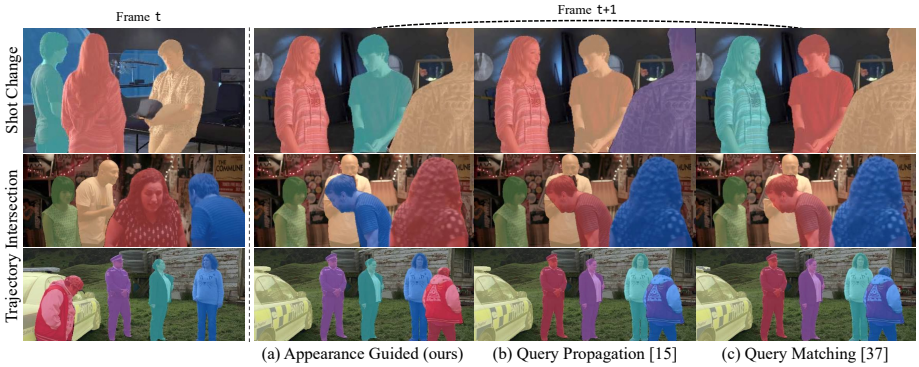

Fig. 1: Qualitative results across challenging scenarios. Predicted results using query-propagation [15, 38], query-matching [17, 31, 37], and our appearance-guided methods. The first row illustrates a shot change across consecutive frames, a scenario where previous methods fail to maintain consistent tracking. The second and third rows demonstrate trajectory intersections, leading to id-switching with previous methods. Unlike previous methods, our method successfully tracks objects without switching or losses. Best viewed in color.

propagation [15, 23] or matching [17, 31, 37] strategies. However, tracking under complex scenarios such as *shot changes* or *trajectory intersections* (Fig. 1) remains imperfect, resulting in the degradation of the overall accuracy.

Examining these failure cases, we observe that *object-wise information* of the queries is significantly imbalanced: heavy reliance on positional cues, and less reflection on appearances. As demonstrated in Fig. 1, previous query-based VIS methods [15, 17, 31, 37, 38], tend to maintain the spatial order of previous predictions in their current predictions. To support this argument, we conduct additional analyses by horizontally flipping images to generate two-frame pseudo videos. The existing models manifest association errors despite the distinct exterior patterns of objects, as shown at the top of Fig. 2, which highlight the dependence on object locations. As there exist multiple scenarios that cannot be fully handled with the imbalanced information, such a phenomenon necessitates the models to take object appearances into consideration.

We introduce **VISAGE** (**V**ideo **I**nstance **S**egmentation with **A**ppearance-**G**uided **E**nhancement), a method that leverages appearance cues as a crucial indicator for distinguishing instances. In our approach, we introduce a streamlined branch that employs mask pooling to generate *appearance* queries from the predicted mask of *object* queries. This enables each appearance query to capture the visual features of its corresponding object, providing a more comprehensive representation for improved tracking accuracy. To refine query discrimination, we integrate a contrastive loss [3, 8, 27], which enhances the model's ability to distinguish between instances across different frames.

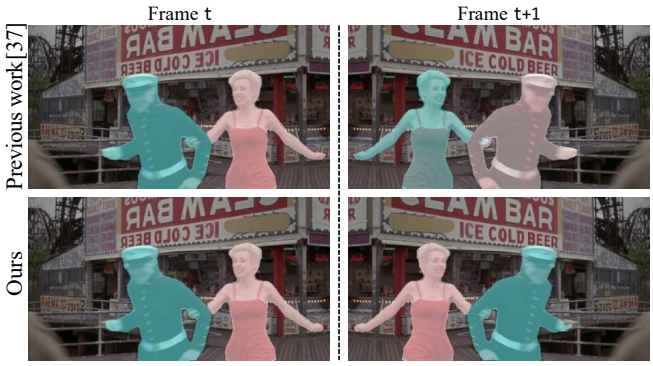

Fig. 2: Proof of concept demonstrated with a flipped image. Previous methods [15, 17, 31, 37, 38] struggle with instance matching in flipped images, showing a dependency on location. Our method, VISAGE, addresses this by emphasizing appearance, enabling accurate instance matching even with image flipping.

Additionally, we introduce a streamlined tracker designed to minimize reliance on heuristic procedures to the greatest extent feasible. Previous methods [31, 37] employ multiple refinement steps for removing redundant mask proposals and adopt handcrafted threshold values for accurate tracklet construction. Through such processing, only selectively chosen queries are incorporated into the matching process. Although these complex tracker configurations enhance tracking accuracy, they also create a dependency on numerous hyperparameters, each of which can be tailored heuristically to specific datasets. To alleviate such dependence, our method streamlines the tracker and dramatically reduces the number of required hyperparameters, such as threshold values for initializing and deleting tracklets, and non-maximum suppression (NMS), among others. Nonetheless, a lack of temporal information still exists, inherent in query-matching online methods, as they are only aware of adjacent frames. We address this limitation by using a simple memory bank to facilitate temporal awareness.

Despite the simplicity of our approach, VISAGE has many desirable properties. Our method introduces a new paradigm in query-based VIS by emphasizing the crucial role of appearance information for object association. Enhanced by appearance-based guidance, it demonstrates superior performance in complex tracking scenarios, outperforming previous methods that often misidentify objects due to an excessive dependence on spatial information as shown in Fig. 1(a). It successfully leverages appearance information, as illustrated at the bottom of Fig. 2, and has been validated on our proposed large-scale pseudo dataset, outperforming established methods [15, 37] by a large margin. Furthermore, with its simplified tracker that effectively utilizes the past history of both object and appearance queries, our method demonstrates competitive performance across all benchmark datasets. Notably, VISAGE achieves state-of-the-art performance on three standard benchmarks: YouTubeVIS-19/21 [32, 33] and OVIS [26].

2 Related Works

2.1 Online Video Instance Segmentation

Contrary to traditional online approaches [11, 19, 32, 35], modern online methods utilize query-based detectors [2, 5, 39] with an emphasis on query association strategies. Query-based online methods can be divided into two main strategies: query-matching and query-propagation.

Query-matching approaches dynamically construct tracklets in an online fashion using query-based detectors [2, 5, 39], which yield predictions for each video frame individually. MinVIS [17] implements this concept by exclusively training of the query-based detector, subsequently deploying it on video frames independently during the inference phase to conduct tracking via bipartite matching of corresponding queries. However, its supervision is restricted to the frame-level, which can introduce ambiguity by not accounting for the object's continuity in the video sequence. To mitigate this, some studies [31, 37] have incorporated contrastive learning [3,8,27] to refine instance embeddings. Moreover, these methods utilize a memory bank at the inference stage, which allows for the processing of multiple frames, thereby enhancing the temporal information captured for each object. Specifically, CTVIS [37] further improves discriminative capability by using the memory bank during the training stage. However, these previous approaches still fall short in discriminative ability due to their insufficient use of appearance information. Our method, VISAGE, introduces a novel strategy that utilizes the appearance cue to significantly enhance discriminative ability, leading to more robust tracking.

Query-propagation methods track objects in video sequences by utilizing output queries from prior frames. By propagating output queries, these methods track corresponding objects across frames [6]. Additionally, some enhance tracking accuracy by also propagating proposals alongside the queries [12]. Recent developments have enabled some methods to operate in both online and offline modes, processing videos clip-by-clip and frame-by-frame. For instance, GenVIS [15] employs an offline VIS method as its backbone, wherein learnable instance prototypes aggregate the backbone's outputs through the propagation of the instance prototype. This design allows GenVIS to function as either an online or offline method, depending on the length of clip processed by the backbone. On the other hand, DVIS [38] constructs tracklets by propagating frame queries in an online manner. These well-established tracklets are then refined to effectively utilize information from the entire video in an offline manner. Yet, these methods are often constrained by a local matching strategy that focuses on aligning tracklets with ground truths based solely on the current frame. This approach frequently leads to unstable tracking outcomes, as it fails to consider the entire video context, resulting in suboptimal performance. TCOVIS [23] addresses this issue by shifting from the local matching strategy of GenVIS [15] to a global matching approach, thereby achieving more robust tracking across videos.

2.2 Offline Video Instance Segmentation

Offline VIS architectures [1, 29] process input videos at the clip-level rather than frame by frame. VisTR [28] extends DETR [2] from frame-level to clip-level processing in an end-to-end manner by simultaneously handling multiple frame features within the transformer encoder-decoder. However, this approach, which processes multiple frame-level inputs at once, demands extensive computation, making the processing of longer sequences impractical.

To overcome this limitation, IFC [18] introduces a memory token in the transformer encoder and employs fixed-size clip queries in the transformer decoder, enhancing both the model's performance and efficiency. With the transformer decoder design of IFC, Mask2Former-VIS [4] adapts Mask2Former [5] for video-level tasks, resulting in substantial performance gains. Meanwhile, SeqFormer [30] redesigns the transformer decoder to process each frame individually, thus improving the model's ability to detect instance movement by precisely capturing location changes. VITA [16] introduces a novel strategy by processing clip queries through cross-attention with frame queries rather than relying on frame features. This strategy lightens the computational load imposed by dense spatio-temporal backbones, resulting in an efficient architecture capable of managing lengthy videos. Beyond focusing solely on transformer architecture enhancements, TeViT [36] introduces a groundbreaking backbone that improves temporal information processing by capitalizing on the strengths of ViT [7].

3 Method

3.1 Query-based Detector

Query-based object detectors [2, 5, 39] can be largely divided into three components: backbone, transformer encoder, and transformer decoder. The backbone initiates the process by generating low-level image feature maps, encapsulating essential visual information. These feature maps are then enhanced through the transformer encoder, which employs self-attention mechanisms to refine the feature representation, as detailed in [2, 39]. The process concludes in the transformer decoder, where the identified objects are decoded into N learnable queries. Our approach adopts the well-established query-based detector framework [5], maintaining its original structure intact.

3.2 Appearance-Guided Enhancement

Prior to the advent of query-based tracking approaches, traditional video tracking methods [8, 20, 32] extracted instance features from backbone feature maps using operations such as RoIPool [10] and RoIAlign [13]. Following a similar principle, we use average pooling to extract appearance queries from the backbone feature maps, guided by the predicted masks as shown in Fig. 3 (a). These appearance queries are designed to encapsulate appearance-centric features, providing a distinctive complement to the object queries. Consequently, alongside the object

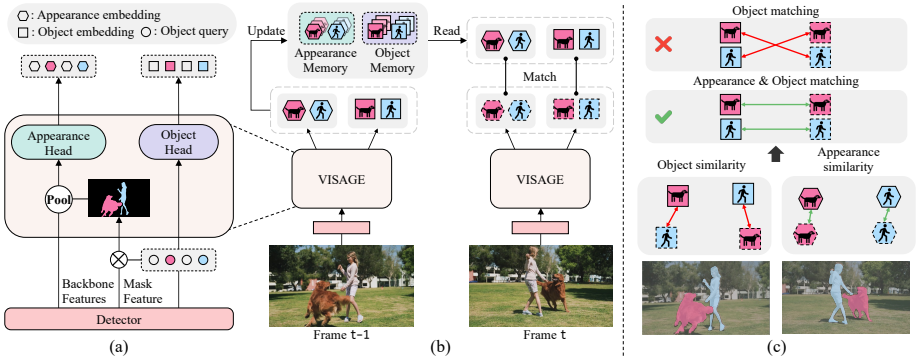

Fig. 3: Overview of VISAGE. (a) The proposed VISAGE's architecture which generate object embedding and appearance embedding. (b) Overall inference pipeline of VISAGE: At time step $t-1$, the memory bank is updated with both the appearance embedding and the object embedding. Then, at time step t, the memory embedding is read from the memory bank and used for matching. (c). Details of the matching process: In that scenario, using only object embeddings leads to incorrect matching. However, when guided by the appearance embedding, the matching process can be corrected. Best viewed in color.

queries already in use, we introduce appearance queries as an additional indicator. We then transform both types of queries into *appearance embeddings* $\mathbf{e}_a \in \mathbb{R}^{N\times C}$ and *object embeddings* $\mathbf{e}_i \in \mathbb{R}^{N\times C}$, respectively.

Our appearance embeddings enhance the matching process when using only object embeddings leads to incorrect matches, as illustrated in Fig. 3 (c). Relying solely on the similarity of object embeddings may lead to ambiguity due to their similar positions across subsequent frames. However, this ambiguity can be resolved by also considering the similarity of appearance embeddings, as their distinct appearances provide additional discriminative information. With this guidance, we leverage both appearance embeddings $\mathbf{e}_a$ and object embeddings $\mathbf{e}_i$ to identify the optimal match between queries across subsequent frames as shown in Algorithm 1 (lines 4-10).

To enhance object association quality, we improve the distinctiveness of both object and appearance embeddings. We utilize contrastive learning to refine embeddings obtained from two distinct frames, ensuring that embeddings of identical object instances are brought closer together in the embedding space, while those of different instances are separated further apart. Unlike previous methods utilizing contrastive learning [8, 31], our approach treats object and appearance embeddings individually, applying contrastive loss to each respectively. It allows each type of embedding to be distinctly characterized by its inherent properties, facilitating their mutual synergy in object matching, as illustrated in Fig. 3 (c). Consequently, our model's final loss integrates a weighted sum of

the contrastive losses for both types of embeddings with the original query-based detector's loss.

Algorithm 1 Inference pipeline of VISAGE.

Input: model $\mathcal{F}$, frames $\{x^t\}_{t=1}^{T}$, weight α
Output: predictions $\mathcal{P}$

$\mathcal{P} \leftarrow \{\}$
$\mathcal{M} \leftarrow \{\}$
$\texttt{idx} \leftarrow [0, N-1]$
for t in $[1, T]$**:**
  $\mathbf{p}^t, \mathbf{e}_i^t, \mathbf{e}_a^t \leftarrow \mathcal{F}(x^t)$
  $\mathcal{M} \leftarrow \mathcal{M} + (\mathbf{e}_i^t[\texttt{idx}], \mathbf{e}_a^t[\texttt{idx}])$
  $\mathbf{m}_i^t, \mathbf{m}_a^t \leftarrow \mathcal{M}.\texttt{read\_memory()}$
  $\mathcal{P} \leftarrow \mathcal{P} + \mathbf{p}^t[\texttt{idx}]$
  $\mathbf{s} \leftarrow (1-\alpha) \cdot \texttt{cos\_sim}(\mathbf{e}_i^t, \mathbf{m}_i^t)$
    $+ \alpha \cdot \texttt{cos\_sim}(\mathbf{e}_a^t, \mathbf{m}_a^t)$
  $\texttt{idx} \leftarrow \texttt{linear\_sum\_assignment}(\mathbf{s})$
end for
return $\mathcal{P}$

3.3 Inference with Appearance

As shown in Algorithm 1, our tracker employs a simple yet effective matching process that aligns the current object embeddings and appearance embeddings with their respective counterparts from the previous frame. We compute the similarity scores for each object-appearance embedding pair using Cosine Similarity, following a method similar to that in [17]. Then, employing the weighted sum of object and appearance similarities, we utilize the Hungarian algorithm [21] (`linear_sum_assignment` in Algorithm 1) to achieve optimal assignment.

In addition, we incorporate a simple memory bank to compensate for the lack of temporal information, a limitation inherent in online processes, as used in previous methods [31, 37]. We stack the states of previous queries from the most recent frames within our memory bank, which has a size of W. From this bank, we read a memory embedding $m \in \mathbb{R}^{N \times C}$ using the `read_memory()` function in Algorithm 1.

Specifically, this function is implemented by temporally weighting the embeddings to put more emphasis on recent queries while utilizing the confidence scores for selective weighting. For the current memory embedding m^t, the calculation of the weighting for each embedding at the previous time step $w \in [1, W]$ can be formally expressed as follows:

$$\mathbf{m}^t = \sum_{w=1}^{W} \left(\mathbf{e}^{t-w} s^{t-w} \times \frac{W}{w} \right), \tag{1}$$

where s denotes the confidence score. This memory embedding represents the object and appearance, denoted as m_i^t and m_a^t respectively, in Algorithm 1.

Finally, we sequentially associate the predictions (**p** in Algorithm 1) from each frame using the obtained assignment. By employing a simple inference pipeline coupled with an efficient memory bank, we introduce an effective approach.

4 Experiments

4.1 Datasets

We evaluate VISAGE on three VIS benchmarks: Youtube-VIS (YTVIS) 2019 / 2021 [32, 33] and Occluded VIS (OVIS) [26]. The YTVIS datasets contain 40 predefined categories in their videos. YTVIS 2019 is the first and largest dataset for video instance segmentation. It includes 2,238 videos for training, 302 for validation, and 343 for the testing. YTVIS 2021 expands upon the YTVIS 2019 dataset, containing 2,985 videos for training, 421 for validation, and 453 for testing, while refining annotations and modifying some categories.

OVIS stands out for its more complex and longer videos, featuring 25 categories and comprising 607 videos for training, 140 for validation, and 154 for testing. In comparison to YTVIS, it contains a higher number of instances per video, averaging 5.8, and a total of 296k masks. Additionally, the average length of videos in OVIS is approximately 12 seconds.

4.2 Implementation Details

We adopt the Mask2Former [5] as our query-based detector. All of our models are initialized with parameters pre-trained on COCO dataset [24] with ResNet-50 [14] backbone. We also adopt COCO joint training following previous works [15, 16, 23, 30, 37]. Our batch includes 16 videos. In our experimental setup, we set the weights for the contrastive losses applied to both the appearance and object embeddings at 2.0. For losses other than the contrastive loss, we adopt the same loss function and weight specifications as those described in [4]. The window size of memory bank W is set to 5 and appearance weight α is set to 0.75 during inference. Finally, our method is trained using 4 NVIDIA A6000 GPUs.

4.3 Main Results

We compare our methods with state-of-the-art methods on three VIS benchmarks: YTVIS 2019/2021 and OVIS. The results are reported in Tab. 1.

Youtube-VIS 2019 & 2021. As shown in Tab. 1, we compare our VISAGE with previous state-of-the-art methods. On the YTVIS 2019 benchmark, VISAGE performs on par with the highest-performing method cited as CTVIS [37]. Notably, both VISAGE and CTVIS [37] surpass previous methods by a large margin. On the YTVIS 2021 benchmark, an improved version of YTVIS 2019, VISAGE outperforms other existing methods. We achieve the highest performance on both the YTVIS 2019 and 2021 datasets by incorporating appearance-guided enhancement.

Table 1: Comparisons on the **YouTube-VIS 2019, 2021 and OVIS validation** sets. Methods are denoted as online or offline, indicated by the text color. We highlight the best performance in **bold**.

Method	Setting	YouTube-VIS 2019					YouTube-VIS 2021					OVIS				
		AP	AP_{50}	AP_{75}	AR_1	AR_{10}	AP	AP_{50}	AP_{75}	AR_1	AR_{10}	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
IFC [18]	offline	41.2	65.1	44.6	42.3	49.6	35.2	55.9	37.7	32.6	42.9	-	-	-	-	-
Mask2Former-VIS [4]	offline	46.4	68.0	50.0	-	-	40.6	60.9	41.8	-	-	-	-	-	-	-
MinVIS [17]	online	47.4	69.0	52.1	45.7	55.7	44.2	66.0	48.1	39.2	51.7	25.0	45.5	24.0	13.9	29.7
IDOL [31]	online	49.5	74.0	52.9	47.7	58.7	43.9	68.0	49.6	38.0	50.9	30.2	51.3	30.0	15.0	37.5
VITA [16]	offline	49.8	72.6	54.5	49.4	61.0	45.7	67.4	49.5	40.9	53.6	19.6	41.2	17.4	11.7	26.0
GenVIS [15]	online	50.0	71.5	54.6	49.5	59.7	47.1	67.5	51.5	41.6	54.7	35.8	**60.8**	36.2	16.3	39.6
GenVIS [15]	offline	51.3	72.0	57.8	49.5	60.0	46.3	67.0	50.2	40.6	53.2	34.5	59.4	35.0	16.6	38.3
DVIS [38]	online	51.2	73.8	57.1	47.2	59.3	46.4	68.4	49.6	39.7	53.5	31.0	54.8	31.9	15.2	37.6
DVIS [38]	offline	52.6	76.5	58.2	47.4	60.4	47.4	71.0	51.6	39.9	55.2	34.1	59.8	32.3	15.9	41.1
TCOVIS [23]	online	52.3	73.5	57.6	49.8	60.2	49.5	71.2	53.8	41.3	55.9	35.3	60.7	**36.6**	15.7	39.5
CTVIS [37]	online	**55.1**	**78.2**	59.1	**51.9**	**63.2**	50.1	73.7	54.7	41.8	**59.5**	35.5	**60.8**	34.9	16.1	**41.9**
VISAGE	online	**55.1**	78.1	**60.6**	51.0	62.3	**51.6**	**73.8**	**56.1**	**43.6**	59.3	**36.2**	60.3	35.3	**17.0**	40.3

OVIS. In Tab. 1, we present a comparison on the OVIS benchmark [25], which is characterized by long videos and complex scenarios, including frequent occlusions. VISAGE achieves performance on par with the previously established state-of-the-art methods such as GenVIS [15], TCOVIS [23], and CTVIS [37]. Through our proposed properties, which include appearance-guided enhancement and a simplified tracker, VISAGE effectively handles such long and complicated videos. As a result, these advancements enable VISAGE to achieve state-of-the-art performance.

4.4 Ablation Studies

In this section, we provide the ablation studies for our proposed method and discuss its effect. All ablation experiments are conducted on YTVIS 2019 [32] validation set.

Appearance feature. In Tab. 2, we analyze the impact of feature maps which create the appearance query. When we make the appearance query from the transformer encoder feature maps, there is a degradation in performance. This indicates that backbone feature maps contain rich visual information compared to transformer encoder feature maps.

Appearance guidance. Tab. 3 shows effectiveness of our appearance-guided enhancement. We omit appearance information by setting the α in line 9 of Algorithm 1 to 0, resulting in a matching process that relies solely on object similarity. As indicated by rows 1 and 3 in Tab. 3, the absence of appearance information leads to a degradation in performance. Further analysis of the effectiveness of appearance guidance is discussed in Sec. 4.5.

Table 2: Ablation study of the target feature maps which generate appearance query

Feature	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
Transformer Encoder	51.4	72.4	56.7	49.8	60.7
Backbone	55.1	78.1	60.6	51.0	62.3

Table 3: Ablation study on appearance guidance and memory bank utilization, with memory bank window size $W = 5$.

App	Memory Bank	YouTube-VIS 2019 AP	AP_{50}	AP_{75}	AR_1	AR_{10}
		49.9	71.4	54.7	47.0	58.7
✓		50.2	72.1	54.7	47.3	60.7
	✓	53.4	76.8	58.7	49.8	61.2
✓	✓	55.1	78.1	60.6	51.0	62.3

Table 4: Ablation study of the tracklet post-processing.

Tracker	NMS	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
[31, 37]		52.7	75.3	57.6	49.3	61.6
[31, 37]	✓	54.4	77.9	59.0	49.9	62.8
ours		55.1	78.1	60.6	51.0	62.3

Memory bank. In Tab. 3, we demonstrate the necessity of the memory bank for compensating for the lack of temporal information. We evaluate performance differences between scenarios with and without the use of the memory bank. Rows 1 and 2 in Tab. 3 show decreased performance compared to rows 3 and 4, respectively. Without a memory bank, our method is constrained to using only the immediately preceding frame for historical context, relying on similarities measured exclusively between consecutive frames. However, the introduction of a memory bank expands this capability by leveraging a broader historical perspective. Furthermore, the adoption of a memory bank significantly boosts the effectiveness of appearance information. This is because historical appearance information provides a more reliable basis for matching than only considering the appearance from the immediate preceding frame to recognize an identical object. By implementing a memory bank, our method gains awareness of previous frames, which leads to improved performance.

Tracklet processing. Previous query-matching based methods [31, 37], build upon the tracker framework from [8], incorporating handcrafted thresholding and heuristic post-processing techniques, including Non-Maximum Suppression (NMS). In contrast, our tracker exclusively utilizes cosine similarity and the Hungarian algorithm for matching, as detailed in Algorithm 1. To understand the impact of handcrafted designs, we align our inference pipeline with those of methods previously used [31, 37]. The results, as detailed in Tab. 4, reveal that our streamlined approach consistently surpasses the performance of the conventional tracker. A detailed examination, especially between rows 1 and 2 of Tab. 4, reveals that the absence of heuristic elements, such as NMS, leads to a decline in performance. However, row 3 reveals that our simplified tracker performs impressively even without heuristic design elements. This suggests that

Table 5: Analysis of Appearance Weight α. The plot demonstrates the relationship between different appearance weight α settings and their corresponding AP scores on the YouTube-VIS 2019 and OVIS validation sets.

α	YouTube-VIS 2019					OVIS				
	AP	AP_{50}	AP_{75}	AR_1	AR_{10}	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
0.00	53.4	76.8	58.7	49.8	61.2	32.2	55.6	30.9	16.1	36.3
0.25	53.6	77.0	59.4	50.0	61.1	34.5	59.2	32.5	16.7	38.8
0.50	54.5	77.2	60.3	50.9	61.7	34.8	59.8	35.3	16.2	39.4
0.75	55.1	78.1	60.6	51.0	62.3	36.2	60.3	35.3	17.0	40.3
1.00	24.9	36.7	28.7	40.9	53.7	11.4	22.7	10.2	10.3	24.9

a simplified tracker is capable of achieving commendable performance without the complexity of heuristic design elements.

4.5 Analysis of Appearance-Guided Enhancement

Appearance Weight. In our methodology, as detailed in line 9 of Algorithm 1, the hyperparameter α plays a crucial role by representing the weighting of the appearance similarity. By tuning the value of α, we modulate the emphasis placed on the appearance information. In Tab. 3, we demonstrate how our appearance guidance significantly enhances performance on the YTVIS 2019 dataset. In this section, we further analysis the impact of appearance cue on robust matching by exploring various values of α.

Tab. 5 highlights the critical role of both appearance and location information in tracking. We analyze the impact of appearance weight α on both the YTVIS 2019 and OVIS validation sets. There are two extreme cases to consider: relying solely on location information or exclusively on appearance information. We observe a reduced performance on both datasets when the α value is set to 0, with this reduction being more pronounced on the OVIS dataset. Given OVIS's complex scenarios, such as frequent occlusions, the dataset's intricacies make the contribution of appearance information especially significant. Conversely, setting α to 1, and thus relying only on appearance information, results in a significant drop in performance. Without the positional cue, the model only depends on appearance information for matching across frames, leading to potential ambiguities in making object tracklets.

However, integrating both appearance and location information consistently surpasses these two extreme cases, highlighting the complementary strengths of using both cues in establishing robust tracklets. Notably, increasing the value of α, and thereby the emphasis on appearance information, correlates with performance enhancements.

T-SNE Visualization. As illustrated in Fig. 4, we analyze the impact of appearance-guided enhancement in VISAGE on its association capabilities using

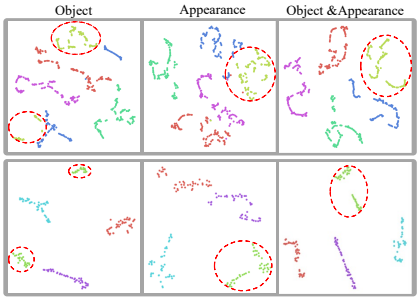

Fig. 4: T-SNE visualization on the OVIS dataset. Each row representing three different videos. Each column corresponds to the type of query embedding utilized. Points plotted in the same color indicate the same instance across the dataset. Best viewed in color.

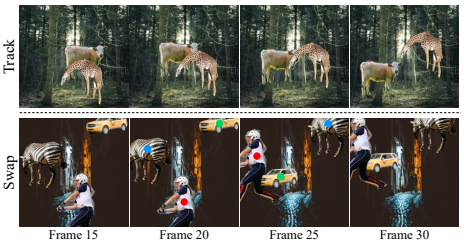

Fig. 5: Visualization of the pseudo dataset. In *track type* videos, instances move along random bezier curves. On the other hand, the *swap type* refers to a scenario where the positions of each instance are exchanged in the middle of the video. The colored dot above each instance represents the corresponding instance in the swapped frame.

the OVIS dataset by visualizing the query embeddings for an entire video, with consistent colors indicating the same instance. The visualization includes three types of query embeddings: object-only, appearance-only, and a combination of both object and appearance queries. While object-only embeddings for the same object exhibit scattered and inconsistent clustering, appearance-only query embeddings demonstrate a high degree of clustering. The integration of object and appearance query embeddings results in even more distinct and pronounced clustering, as exemplified by the red circle in Fig. 4. This enhanced clustering clearly indicates that our appearance-guided enhancement leads to better association.

Pseudo Dataset. Traditional datasets do not fully cover the complex scenarios that our approach is designed to address. To validate VISAGE in a more intuitive manner, we construct a pseudo dataset consisting of synthetic videos. Instances in pseudo videos are created by compositing objects from the COCO dataset [24] with Copy-and-Paste augmentation [9]. The background images for the pseudo dataset are randomly sourced from BG-20k [22]. Additionally, we employ Bezier Curves to simulate the movement of objects. It includes two types of videos: *track*, where objects move following arbitrary Bezier Curves, and *swap*, where objects' locations are randomly swapped along their trajectories. Except for the movement of instances, both types of datasets are generated under the same conditions.

As shown at the top of Fig. 5, instances in the track type pseudo video move along a Bezier Curve. Consequently, complex scenarios, such as intersections between instances or movements out of the frame, occur naturally. On the other hand, the swap type presents a challenge for methods that primarily rely on location information, as this dependence results in incorrect matching. This is

Table 6: Comparisons on **Pseudo dataset**.

	Method	App	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
Track	GenVIS [15]		54.8	70.6	58.8	61.2	65.9
	CTVIS [37]		63.7	78.7	69.7	70.3	74.8
	VISAGE		65.2	**80.9**	71.3	70.2	75.2
	VISAGE	✓	**65.6**	80.7	**71.8**	**70.5**	**75.8**
Swap	GenVIS [15]		41.8	59.7	44.0	50.8	57.0
	CTVIS [37]		53.1	71.7	56.9	61.7	65.7
	VISAGE		51.8	72.1	54.4	57.7	62.8
	VISAGE	✓	**66.1**	**81.8**	**73.3**	**69.9**	**75.6**

illustrated at the bottom of Fig. 5, where the positions of instances in the pseudo video are suddenly swapped. Such scenarios verify the method's awareness of appearance cues when location information is no longer a reliable indicator.

In Tab. 6, we conduct an evaluation of various online VIS methods [15, 37] including VISAGE on the pseudo dataset using published ResNet-50 backbone weights, which are trained on the YouTube-VIS 2019 dataset. All of these methods use COCO joint training, ensuring a fair comparison. The overall trend is similar to that observed on the YTVIS 2019 dataset. In contrast, VISAGE outperforms other methods in swap type videos, mainly due to the challenges posed by the dataset in associating instances using only location information. Consequently, other methods demonstrate degraded performance on swap type videos compared to their performance on track types. Furthermore, the absence of appearance information in the matching process notably impacts VISAGE's performance on swap type videos, although it remains relatively unaffected for track type videos. It proves that appearance cue serves as an additional indicator, providing robustness in scenarios with swapped object positions.

4.6 Qualitative Results

Fig. 6 displays the visualization results of our VISAGE across various challenging scenarios. In the first row, several cats are positioned closely together and the red-colored cat crosses another cat. In such cases, relying solely on location information can often result in identity-switching errors. In the second row, we present a similar scenario involving different classes: a person and a cow.

In the third row, we showcase another challenging scenario where an object completely disappears and then reappears: a cat disappears and then reappears. Our memory bank, which stores the previous queries of the cat, enables us to reestablish the association when the cat reappears. The fourth row demonstrates a similar scenario, where our memory bank proves effective: a small sedan is completely occluded by the truck. Even though the large truck in the foreground and the small, disappeared car are located closely, their distinct appearances ensure accurate matching.

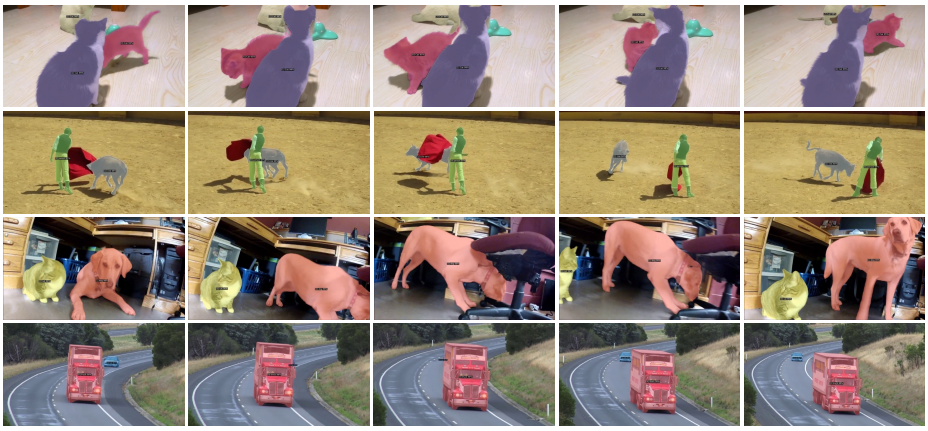

Fig. 6: Qualitative results of VISAGE. Videos are sourced from OVIS [26] and YTVIS 2021 [33] datasets. These videos represent complex scenarios, characterized by intersections and reappearances. Best viewed in color.

5 Limitations and Future Works

VISAGE addresses the lack of appearance awareness in existing query-based online VIS methods. Nevertheless, it faces two main limitations. Firstly, while VISAGE primarily employs a query-matching strategy, we recognize that query-propagation methods also depend on spatial information. This observation suggests a future research direction: integrating appearance information awareness into query-propagation approaches. Secondly, a fundamental issue not unique to our method but prevalent in tracking-by-detection approaches, especially in recent query-based VIS methods, is the heavy reliance on frame-level detectors. During the tracking stage, these methods do not account for their own errors, allowing any inaccuracies to adversely affect video-level predictions. Addressing this challenge requires thorough consideration and represents another avenue for future work.

6 Conclusion

In this paper, we explore the importance of appearance in tracking objects, an aspect often taken for granted yet overlooked by current VIS methods. By leveraging appearance cues, our simple yet effective method achieves comparable performance to previous methods across various VIS benchmarks. However, as existing VIS benchmarks do not focus on appearance-requiring scenarios, we generate a synthetic dataset that necessitates the use of appearance information. This dataset serves to validate our appearance-aware approach and our approach surpasses other methods with very large margin. We believe that recognizing and leveraging the importance of appearance can lead to progress of VIS.

Acknowledgements

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (Artificial Intelligence Graduate School Program, Yonsei University, under Grant 2020-0-01361) and Artificial Intelligence Innovation Hub under Grant RS-2021-II212068.

References

1. Athar, A., Mahadevan, S., Ošep, A., Leal-Taixé, L., Leibe, B.: Stem-seg: Spatio-temporal embeddings for instance segmentation in videos. In: ECCV (2020)
2. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV (2020)
3. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML (2020)
4. Cheng, B., Choudhuri, A., Misra, I., Kirillov, A., Girdhar, R., Schwing, A.G.: Mask2former for video instance segmentation. arXiv preprint arXiv:2112.10764 (2021)
5. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: CVPR (2022)
6. Choudhuri, A., Chowdhary, G., Schwing, A.G.: Context-aware relative object queries to unify video instance and panoptic segmentation. In: CVPR (2023)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
8. Fischer, T., Huang, T., Pang, J., Qiu, L., Chen, H., Darrell, T., Yu, F.: Qdtrack: Quasi-dense similarity learning for appearance-only multiple object tracking. TPAMI (2023)
9. Ghiasi, G., Cui, Y., Srinivas, A., Qian, R., Lin, T.Y., Cubuk, E.D., Le, Q.V., Zoph, B.: Simple copy-paste is a strong data augmentation method for instance segmentation. In: CVPR (2021)
10. Girshick, R.: Fast r-cnn. In: ICCV (2015)
11. Han, S.H., Hwang, S., Oh, S.W., Park, Y., Kim, H., Kim, M.J., Kim, S.J.: Visolo: Grid-based space-time aggregation for efficient online video instance segmentation. In: CVPR (2022)
12. He, F., Zhang, H., Gao, N., Jia, J., Shan, Y., Zhao, X., Huang, K.: Inspro: Propagating instance query and proposal for online video instance segmentation. In: NeurIPS (2022)
13. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask r-cnn. In: ICCV (2017)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
15. Heo, M., Hwang, S., Hyun, J., Kim, H., Oh, S.W., Lee, J.Y., Kim, S.J.: A generalized framework for video instance segmentation. In: CVPR (2023)
16. Heo, M., Hwang, S., Oh, S.W., Lee, J.Y., Kim, S.J.: Vita: Video instance segmentation via object token association. In: NeurIPS (2022)
17. Huang, D.A., Yu, Z., Anandkumar, A.: Minvis: A minimal video instance segmentation framework without video-based training. In: NeurIPS (2022)

18. Hwang, S., Heo, M., Oh, S.W., Kim, S.J.: Video instance segmentation using inter-frame communication transformers. In: NeurIPS (2021)
19. Ke, L., Li, X., Danelljan, M., Tai, Y.W., Tang, C.K., Yu, F.: Prototypical cross-attention networks for multiple object tracking and segmentation. In: NeurIPS (2021)
20. Kim, D., Woo, S., Lee, J.Y., Kweon, I.S.: Video panoptic segmentation. In: CVPR (2020)
21. Kuhn, H.W.: The hungarian method for the assignment problem. NRL (1955)
22. Li, J., Zhang, J., Maybank, S.J., Tao, D.: Bridging composite and real: Towards end-to-end deep image matting. IJCV (2022)
23. Li, J., Yu, B., Rao, Y., Zhou, J., Lu, J.: Tcovis: Temporally consistent online video instance segmentation. In: ICCV (2023)
24. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
25. Qi, J., Gao, Y., Hu, Y., Wang, X., Liu, X., Bai, X., Belongie, S., Yuille, A., Torr, P.H., Bai, S.: Occluded video instance segmentation. arXiv preprint arXiv:2102.01558 (2021)
26. Qi, J., Gao, Y., Hu, Y., Wang, X., Liu, X., Bai, X., Belongie, S., Yuille, A., Torr, P.H., Bai, S.: Occluded video instance segmentation: A benchmark. IJCV (2022)
27. Sohn, K.: Improved deep metric learning with multi-class n-pair loss objective. In: NeurIPS (2016)
28. Wang, Y., Xu, Z., Wang, X., Shen, C., Cheng, B., Shen, H., Xia, H.: End-to-end video instance segmentation with transformers. In: CVPR (2020)
29. Wu, J., Yarram, S., Liang, H., Lan, T., Yuan, J., Eledath, J., Medioni, G.: Efficient video instance segmentation via tracklet query and proposal. In: CVPR (2022)
30. Wu, J., Jiang, Y., Zhang, W., Bai, X., Bai, S.: Seqformer: a frustratingly simple model for video instance segmentation. In: ECCV (2022)
31. Wu, J., Liu, Q., Jiang, Y., Bai, S., Yuille, A., Bai, X.: In defense of online models for video instance segmentation. In: ECCV (2022)
32. Yang, L., Fan, Y., Xu, N.: Video instance segmentation. In: ICCV (2019)
33. Yang, L., Fan, Y., Xu, N.: The 3rd large-scale video object segmentation challenge - video instance segmentation track (2021)
34. Yang, L., Fan, Y., Xu, N.: The 4th large-scale video object segmentation challenge - video instance segmentation track (2022)
35. Yang, S., Fang, Y., Wang, X., Li, Y., Fang, C., Shan, Y., Feng, B., Liu, W.: Crossover learning for fast online video instance segmentation. In: ICCV (2021)
36. Yang, S., Wang, X., Li, Y., Fang, Y., Fang, J., Liu, W., Zhao, X., Shan, Y.: Temporally efficient vision transformer for video instance segmentation. In: CVPR (2022)
37. Ying, K., Zhong, Q., Mao, W., Wang, Z., Chen, H., Wu, L.Y., Liu, Y., Fan, C., Zhuge, Y., Shen, C.: Ctvis: Consistent training for online video instance segmentation. In: ICCV (2023)
38. Zhang, T., Tian, X., Wu, Y., Ji, S., Wang, X., Zhang, Y., Wan, P.: Dvis: Decoupled video instance segmentation framework. In: ICCV (2023)
39. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. In: ICLR (2021)