Video-kMaX: A Simple Unified Approach for Online and Near-Online Video Panoptic Segmentation

Inkyu Shin1,2† Dahun Kim2 Qihang Yu2† Jun Xie2 Hong-Seok Kim2 Bradley Green2 In So Kweon1 Kuk-Jin Yoon1 Liang-Chieh Chen2†
1KAIST 2Google Research

Abstract

Video Panoptic Segmentation (VPS) aims to achieve comprehensive pixel-level scene understanding by segmenting all pixels and associating objects in a video. Current solutions can be categorized into online and near-online approaches. Evolving over the time, each category has its own specialized designs, making it nontrivial to adapt models between different categories. To alleviate the discrepancy, in this work, we propose a unified approach for online and near-online VPS. The meta architecture of the proposed Video-kMaX consists of two components: within-clip segmenter (for clip-level segmentation) and cross-clip associator (for association beyond clips). We propose clip-kMaX (clip k-means mask transformer) and LA-MB (location-aware memory buffer) to instantiate the segmenter and associator, respectively. Our general formulation includes the online scenario as a special case by adopting clip length of one. Without bells and whistles, Video-kMaX sets a new state-of-the-art on KITTI-STEP and VIPSeg for video panoptic segmentation Code and models are available at this link.

1. Introduction

Video Panoptic Segmentation (VPS) [19] aims at a holistic video understanding of the scene by unifying two critical and challenging tasks: semantically segmenting images and associating segmented regions across all frames in a video [42]. It can benefit various real-world applications, such as autonomous driving, robot visual control, and video editing.

With the rapid growth of interest, there have been several methods [19, 20, 24, 32, 43, 57] proposed for VPS. They can be categorized into online and near-online approaches, which process the video either frame-by-frame or clip-by-clip (a clip contains only a few consecutive video frames). The online approaches, such as VPSNet [19] and Video K-

† Work done while at Google Research.
line models to near-online (and vice versa). Particularly, the current online methods [19, 24] lack a proper clip-level segmenter, while the modern near-online methods [20, 32] fail to associate objects in an online manner, suffering from the absence of overlapping frames. The need for this scenario-specific design results in inefficiencies, as it requires the utilization of different frameworks for each setting. A natural question thus emerges: Is it possible to develop a unified framework for online and near-online VPS without any scenario-specific design?

To answer the question, we carefully design Video-kMaX, a simple yet effective approach for both online and near-online VPS. As drawn in Fig. 1, the meta architecture of Video-kMaX contains two components: within-clip segmenter and cross-clip associater, where the former component performs clip-level segmentation and the later one associates detected objects across clips. The proposed Video-kMaX is an instantiation of the pipeline by adopting clip-kMaX (clip k-means mask transformer) for the within-clip segmenter, and LA-MB (Location-Aware Memory Buffer) for the cross-clip associater. 

The proposed clip-kMaX extends the image-level k-means mask transformer [52] to the clip-level without adding any extra modules or loss functions. Motivated by the k-means clustering perspective [51], we consider object queries as cluster centers, where each query is responsible for grouping pixels of the same object within a clip together. Specifically, each object query, when multiplied with the clip features [35, 38, 40], is learned to yield a tube prediction (i.e., masks of the same object in a clip) [20]. This learning can be achieved via a surprisingly simple modification in the k-means cross-attention module [52] by concatenating the clip-level pixel features along the spatial dimension. As a result, clip-kMaX can be applied to both near-online and online settings without additional complexities. We also empirically show that k-means cross-attention is an effective mechanism for handling the extremely long sequence of spatially and temporally flattened clip features.

The proposed LA-MB is motivated by the drawbacks of existing methods through the careful systematical studies. We observe that the modern VPS methods [19, 32] could not handle the more challenging setting of long-term object tracking, since they either associate objects in the neighboring frames [19] or stitch overlapping frame predictions [32], making it hard to track objects beyond the short clip length. One promising solution is to exploit a memory buffer to propagate the tracking information across all video clips, which has been proven successful in the recent works [16, 45, 49, 53]. However, surprisingly, we observe that naively adopting the memory buffer to VPS leads to minor improvements or even worse performance. The setback enforces us to further look into its root case. We discover that the appearance feature alone [19, 45] is not sufficient for long-term association in VPS, when the target object is occluded for a long time; additionally, the memory buffer approach accumulates many detected objects, resulting in a huge matching space (between stored and newly detected objects) and hindering the matching accuracy. To resolve the issues, we develop LA-MB (Location-Aware Memory Buffer), which effectively incorporates location information to the memory module by two means. First, when comparing the similarity between the stored objects in memory and the detected objects in the current frame, we consider not only their appearance features (encoded by object queries), but also their location features (encoded by normalized bounding box coordinates). Specifically, if the object of interest is not detected in the current frame but it is stored in the memory (e.g., due to occlusion), we will “predict” its current location by assuming the object is moving at a constant velocity. Second, we propose a hierarchical matching scheme to effectively reduce the matching space. We initially exploit the matching results from the Video Stitching [32] strategy, which associates objects based on their mask IoU in the overlapping frame between clips, effective for short-term association. We then associate the objects stored in memory with the currently detected but unmatched objects, aiming for long-term association. Thanks to our careful design, the LA-MB improves the long-term association quality both in near-online and online scenarios with low sensitivity to the hyper-parameter values.

In summary, we introduce Video-kMaX, a simple and unified method for online and near-online VPS. Our approach, consisting of two seamless modules: clip-kMaX and LA-MB, achieves significant performance improvements on two long sequence VPS datasets: KITTI-STEP [42] and VIPSeg [30]. In particular, as shown in Fig. 2, our best Video-kMaX outperforms the previous state-of-the-art online model (Video K-Net [24]) and near-online model (TubeFormer [20]) by +2.5% STQ and +4.6% STQ, respectively, on KITTI-STEP val set.

2. Related Work

In this regard, we propose a simple unified online and near-online video panoptic segmentation model for long-term association by additionally exploiting the memory buffer [45,53]. Particularly, MOTR [53] proposes a set components: clip-\(k\)MaX (clip \(k\)-means mask transformer) for within-clip segmentation (Sec. 3.1) and LA-MB (location-aware memory buffer) for cross-clip association (Sec. 3.2). We detail them below, starting from the near-online framework. Our general formulation includes the online scenario by using clip length one (Sec. 3.3).

3. Method

The meta architecture of Video-\(k\)MaX contains two components: clip-\(k\)MaX (clip \(k\)-means mask transformer) for within-clip segmentation (Sec. 3.1) and LA-MB (location-aware memory buffer) for cross-clip association (Sec. 3.2). We detail them below, starting from the near-online framework. Our general formulation includes the online scenario by using clip length one (Sec. 3.3).

3.1. Within-Clip Segmenter: clip-\(k\)MaX

We first present the general formulation for image and video panoptic segmentation, before introducing our within-clip segmenter clip-\(k\)MaX, which performs clip-level segmentation with a short length \(T\) (e.g., \(T = 2\)).

General Formulation for Image and Video Recently, image panoptic segmentation has been reformulated as a simple set prediction powered by Transformer [36]. From the pioneering works (e.g., DETR [3] and MaX-DeepLab [38]) to the recent state-of-the-art methods (e.g., \(k\)MaX-DeepLab [52]), panoptic predictions are designed to match the ground truth masks by segmenting image \(I \in \mathbb{R}^{H \times W \times 3}\) into a fixed-size set of \(N\) class-labeled masks:

\[
\{\hat{y}_i\}_{i=1}^{N} = \{\hat{m}_i, \hat{p}_i(c)\}_{i=1}^{N},
\]

where \(\hat{m}_i \in [0, 1]^{H \times W}\) and \(\hat{p}_i(c)\) denote predicted mask and semantic class probability for the corresponding mask.
respectively. Motivated by this, TubeFormer [20] extends this formulation into set prediction of class-labeled tubes: \( \{\hat{y}_i\}_{i=1}^N = \{(\hat{m}_i, \hat{p}_i(c))\}_{i=1}^N \), where \( \hat{m}_i \in [0,1]^{T \times H \times W} \). In this setting, \( N \) object queries attend to the \( T \times H \times W \) clip features, and predict \( N \) tubes. The prediction generalizes well for different values of \( T \), since the positional embedding is only performed in the frame level, providing a useful structural prior that the same object in neighboring frames (assuming slow motion) will still be assigned by the same object query. Given the generalizability, we are able to absorb the \( T \)-axis into the \( H \)-axis before feeding the clip features to transformer decoder. Specifically, we propose to relax Eq. (1) into a more general form: \( \{\hat{y}_i\}_{i=1}^N = \{(\hat{m}_i, \hat{p}_i(c))\}_{i=1}^N \), where \( \hat{m}_i \in [0,1]^{S \times W} \), \( S=TH \), and \( T \geq 1 \) (i.e., \( S \) can change according to the different number of frames \( T \)). By doing so, it allows us to easily extend an image panoptic segmentation model to the video domain (clip-level), as detailed below.

**clip-kMax** The state-of-the-art image segmentation model kMaX-DeepLab [52] replaces the cross-attention in a typical transformer decoder [36] with \( k \)-means cross-attention by taking a cluster-wise argmax as below:

\[
\hat{C} = C + \arg \max_N Q^c \times (K^p)^T \times V^p, \tag{2}
\]

where \( C \in \mathbb{R}^{N \times D} \) refers to \( N \) object queries with \( D \) channels. We use superscripts \( p \) and \( c \) to indicate the feature projected from the pixel features and object queries, respectively. \( Q^c \in \mathbb{R}^{N \times D}, K^p \in \mathbb{R}^{HW \times D}, V^p \in \mathbb{R}^{HW \times D} \) stand for the linearly projected features for query, key, and value, respectively. In this \( k \)-means perspective, one object query is regarded as a cluster center, which learns to group pixels of the same object together. Given our previous general formulation, we can seamlessly extend kMaX-DeepLab to video clip, forming our clip-kMaX, by simply reshaping the key and value into \( K^p \in \mathbb{R}^{SW \times D} \) and \( V^p \in \mathbb{R}^{SW \times D} \) (\( S=TH \) and \( T\geq 1 \)). The reshaping merges the \( T \)-frame feature to a \( T \)-frame feature with large height \( TH \) (i.e., reshape \( T \times H \times W \) to \( 1 \times TH \times W \)), which then becomes compatible with the image model kMaX-DeepLab. This is equivalent to performing the \( k \)-means clustering for a video clip with length \( T \), where one query is now learning to group pixels of the same object in the clip together. We illustrate clip-kMaX in Fig. 3. Note that kMaX-DeepLab then becomes a special case of clip-kMaX with \( T = 1 \).

**Discussion** The design of clip-kMaX may look simple on the surface. However, we made strenuous efforts in enhancing the conventional cross-attention module for clip-level mask predictions during its development. When dealing with the extremely large sequence length of spatially and temporally flattened clip features in a video clip, the standard cross-attention module is susceptible to learning. This phenomenon was evident in the poor performance of the original cross-attention, motivating the prior art TubeFormer [20] to further employ an additional latent memory module. To address this challenge, we propose using the \( k \)-means cross-attention [52] approach, which is capable of handling flattened clip features of any size by performing a cluster-wise argmax on \( N \) cluster centers.

**Video Stitching (VS)** In practice, given the limited memory, we are only able to perform clip-level inference (i.e., segmenting a short clip with length \( T \)). To obtain the video-level segmentation, some heuristic designs are required. One popular approach is Video Stitching (VS) [20, 32], which propagates object identities between clips by matching the mask IoU scores in the overlapping frames. In our framework, we adopt the same video stitching strategy for our near-online Video-kMaX, but additionally explore memory buffer for long-term association.

### 3.2. Cross-Clip Associator: LA-MB

Our LA-MB basically consists of two phases: Encoding Phase to store the previous object features, and Decoding Phase to associate current objects with the objects stored in the memory buffer. We detail the process below.

**Encoding Phase** The memory buffer is initially empty, when a new testing video comes. It encodes features from all detected objects, while processing frames sequentially. Regarding the object features to be stored, we exploit the appearance and location properties of each object.

For the appearance feature of object \( i \) observed at frame \( t \), we utilize the query embedding \( q^i_t \in \mathbb{R}^D \) (i.e., object queries from the mask transformer decoder [52]). The memory buffer encodes appearance feature \( q^i_t \) as follows:

\[
\hat{q}^i_t = \begin{cases} (1 - \lambda)q^i_{t-1} + \lambda q^i_t, & \text{if } \text{both in memory and frame } t, \\ q^i_t, & \text{else if } \text{only in frame } t, \\ q^i_{t-1}, & \text{else if } \text{only in memory}, \end{cases}
\]

where \( \lambda \) is the moving average weight between the stored appearance feature in memory \( q^i_{t-1} \) and current appearance feature \( q^i_t \). We set \( \lambda \) to 0.8 as the default value.

Unlike other works [19, 45], we additionally exploit the location feature of object \( i \) observed at frame \( t \), using its normalized bounding box (inferred from the predicted mask); \( b^i_t = (x^{tl}_i/w, y^{tl}_i/h, x^{br}_i/w, y^{br}_i/h) \in \mathbb{R}^4 \), where \( (x^{tl}, y^{tl}) \) and \( (x^{br}, y^{br}) \) are the x-y coordinates of top-left and bottom-right corners, and \( w \) and \( h \) denote the bounding box width and height, respectively. The memory buffer then encodes the location features as follows:

\[
\hat{b}^i_t = \begin{cases} b^i_t, & \text{if in frame } t, \\ \hat{b}^i_t + (\hat{b}^i_{t-1} - \hat{b}^i_{t-2}), & \text{else if } \text{only in memory}. \end{cases}
\]

As shown in the equations, if an object is detected, the memory buffer will use its latest normalized bounding box information. If the object \( i \) is not detected but it is stored in the
memory (e.g., due to occlusion), we will “predict” its current location by assuming the object’s moving velocity is constant, i.e., its location is shifted by \( \hat{b}_i^{t-1} - \hat{b}_i^{t-2} \) from its previous stored location \( \hat{b}_i^{t-1} \).

Finally, the memory buffer stores both the appearance and location features \((\hat{q}_i, \hat{b}_i)\) for all \( M \) objects detected until the current frame. In practice, we adopt the memory refreshing strategy [45], where the old objects, whose last appeared frame is \( \tau \) frame behind the current frame, are removed from the memory buffer. We empirically choose the optimal value for \( \tau \) in our experiments.

**Decoding Phase** To specialize the memory buffer approach in our framework, we initially conduct the Video Stitching (VS) for short-term association between clips. Afterwards, we associate the objects stored in memory with the currently detected but unmatched objects, aiming for long-term association. This hierarchical matching mechanism forms our proposed Location-Aware Memory Buffer (LA-MB). Specifically, we compute the similarity function \( f(i, j) \) between the currently unmatched object \( i \) (after VS) and the encoded object \( j \) in the memory as follows:

\[
f(i, j) = e^{-||\hat{b}_i - \hat{b}_j||^2/T} \cdot \cos(\hat{q}_i, \hat{q}_j).
\]

We compute the negative \( L_2 \) distance between two normalized bounding boxes, weighted by a temperature \( T \) for scaling the values between location and appearance similarity. The appearance similarity is measured by the cosine distance. Then, we obtain a similarity matrix \( S \in \mathbb{R}^{M \times N} \) between \( M \) objects in memory and \( N \) detected objects in the current frame. To find the association, we perform Hungarian matching [23] on \( S \). Additionally, to filter out false associations, we only consider the matching with similarity value larger than a confidence threshold \( \alpha \). The unmatched objects in current frame are considered as new objects. The proposed LA-MB is illustrated in Fig. 4.

**Discussion** Our proposed LA-MB is partially inspired by the success of IDOL [45] in video instance segmentation, and memory buffer has been proven effective in several recent works [2,49,53]. However, there are two critical issues, if one naively applies their memory buffer approach to our framework (we name this method as naïve Memory Buffer (naïve-MB) for our baseline). First, the location feature is not exploited, but only the appearance feature. In a dynamic scene, object location plays an important role. The appearance feature becomes less reliable if the target object has been occluded for a long time. Second, the memory size \( M \) keeps growing as time goes by. Even though this issue is slightly alleviated by the memory refreshing strategy, it still results in a large matching space between the stored \( M \) objects in the memory and the currently detected \( N \) objects, which subsequently makes the one-to-one matching harder.

To overcome the issues, our LA-MB proposes a novel formulation to incorporate the location features (Eq. (4) and Eq. (5)), and additionally augments the matching accuracy by performing the Video Stitching (VS) in the beginning of decoding phase, which effectively further reduces the matching space and improves the matching accuracy.
### 3.3. Online Video Panoptic Segmentation

The meta architecture of Video-kMaX enables a general framework for both online and near-online VPS. When processing a clip of length one, our model performs online VPS. Specifically, the model is trained frame-by-frame and evaluated sequentially with the assistance of clip-kMaX’s general formulation. Unlike the near-online setting, we skip the Video Stitching, which becomes infeasible in the online framework. Afterwards, we apply our LA-MB without any further modification.

### 4. Experimental Results

We conduct experiments on two long sequences Video Panoptic Segmentation datasets: KITTI-STEP [42] and VIPSeg [30].

#### 4.1. Datasets

KITTI-STEP [42] is a Video Panoptic Segmentation (VPS) dataset that contains long video sequences with average track length 51 frames and maximum 643 frames, presenting a challenging scenario for long-term association. It contains 19 semantic classes, similar to Cityscapes [10], while only two classes (‘pedestrians’ and ‘cars’) come with tracking IDs. We adopt the Segmentation and Tracking Quality (STQ) as a metric for evaluation.

VIPSeg [30] is a new large-scale Video Panoptic Segmentation (VPS) benchmark providing in-the-wild real-world scenarios with 232 scenes and 124 classes. Among them, 58 classes are annotated with tracking IDs. The average sequence length is 24 frames per video. We adopt the STQ and VPQ [19] metric for evaluation.

#### 4.2. Implementation Details

The proposed Video-kMaX is a unified approach for online and near-online VPS. For the near-online setting, we employ a clip length of two with one overlapping frame between clips. For the online setting, we set clip length to one and remove the video stitching strategy in the pipeline.

We employ two common backbones for both online and near-online settings: ResNet50 [14] and ConvNeXt-L [28]. We also experiment with Axial-ResNet50-B1 [39] backbone to fairly compare with TubeFormer [20]. Our Video-kMaX is built with the official code-base [41]. Closely following the prior works [20, 42], both the near-online and online models employ a specific pre-training protocol for KITTI-STEP and VIPSeg. They all commonly require ImageNet [33] pretrained checkpoint. VIPSeg further requires pre-training models on COCO [27]. For KITTI-STEP, Cityscapes [10] is additionally adopted as a pre-training dataset since they share a similar driving scene and class category. We note that our best backbone ConvNeXt-L [28] in KITTI-STEP uses both COCO [27] and Cityscapes [10] for pre-training model.

#### 4.3. Main Results

KITTI-STEP Tab. 1 summarizes our performance on the KITTI-STEP val and test set results. On the validation set (Tab. 1 (a)), we compare methods in the two categories: online and near-online methods. In the online setting, when using the standard ResNet50 [14], our Video-kMaX (online) outperforms Video K-Net [24] by +2.5% STQ. To further push the envelope, our model, equipped with the modern backbone ConvNeXt-L [28], achieves the new state-of-the-art with 76.5% STQ. In the near-online setting, when using ResNet50, our Video-kMaX (near-online) significantly surpasses Motion-DeepLab [42] by +16.2% STQ. When employing Axial-ResNet50-B1 [39] backbone, Video-kMaX (near-online) also outperforms TubeFormer [20] by +2.8% STQ. Finally, Video-kMaX (near-online) with ConvNeXt-L further sets a new state-of-the-art
performance with 78.9% STQ, significantly outperforming the current best result (TubeFormer with Axial-ResNet50-B3) by +4.6% STQ. We observe the same trend on the test set (Tab. 1 (b)), where our model reaches 68.5% STQ, significantly outperforming the prior arts TubeFormer [20], Video K-Net [24], and Motion-DeepLab [42] by +3.2%, +5.5%, and +16.3% STQ, respectively. Remarkably, our extremely simple model even outperforms the ICCV 2021 Challenge winning entry, UW IPL/ETRI/AIRL [54] by +0.9% STQ, which exploits pseudo labels [4, 59] and adopts an exceedingly complicated system that not only consists of separate tracking, detection, and segmentation modules, but also requires 3D object and depth information.

VIPSeg Tab. 2 (a) summarizes the results on the VIPSeg val set. In the online setting, our Video-kMaX (online) with ResNet50 attains 38.7% STQ / 36.8% VPQ, significantly outperforming the prior art Video K-Net by +5.6% STQ / +10.7% VPQ. Using the ConvNeXt-L backbone, our model advances the state-of-the-art to 49.4% STQ / 49.4% VPQ. In the near-online setting, when using ResNet50, our Video-kMaX (near-online) surpasses Clip-PanoFCN [30] by +8.4% STQ / +15.3% VPQ. When using Axial-ResNet50-B1, Video-kMaX (near-online) outperforms TubeFormer [20] by +6.0% STQ / +17.5% VPQ. Our best setting with ConvNeXt-L backbone further advances the state-of-the-art to 51.7% STQ / 51.9% VPQ, outperforming TubeFormer with Axial-ResNet50-B3 by +10.2% STQ / +20.7% VPQ. We also show the effectiveness of Video-kMaX (near-online) on VIPSeg test set in Tab. 2 (b), where Video-kMaX also sets a new state-of-the-art, outperforming TubeFormer [20] by +8.5% STQ / +18.2% VPQ.

4.4. Ablation Studies

Association Modules Our proposed LA-MB exploits (1) Video Stitching (VS), (2) appearance feature, and (3) location feature, to perform the object association. In Tab. 3, we carefully study the effect of each feature in LA-MB under both near-online and online settings. In the near-online setting (Tab. 3 (a)), when using these three features individually, we discover that both VS and location feature are equally more effective than appearance feature. We note that when using only the appearance, the method becomes the naïve-MB approach, used by other works [16, 45]. Combining all of them leads to our best final setting, while taking out the location feature will degrade the AQ performance most. This study demonstrates that our proposed location feature is the most effective feature among them. In the online setting, since the VS strategy becomes infeasible, we only experiment with the appearance and location features. As shown in Tab. 3 (b), the pure image-based model, which does not exploit any association feature, attains the worst performance. Interestingly, we notice that the appearance feature learned by the ResNet50 [14] is less effective than ConvNeXt-L [28]. When the appearance feature is less effective (e.g., when using ResNet50), it is better to just use the location feature for association. On the other hand, when the appearance feature is sufficiently informative (e.g., when using ConvNeXt-L), the best performance is obtained by using both appearance and location features.

Memory-related Hyper-parameters Our proposed memory module LA-MB contains two hyper-parameters: $\tau$ (for refreshing old objects in the memory buffer) and $\alpha$ (confidence threshold for matching). In Tab. 4, we ablate their effects on our LA-MB and the baseline naïve-MB. As shown in the table, our LA-MB not only performs better, but also is more robust to the hyper-parameter values than naïve-MB. More concretely, when computing the mean and standard deviation (std) for the obtained AQ w.r.t. different $\tau$ and $\alpha$, our LA-MB achieves a mean of 73.4 and a std of 0.4, while the baseline naïve-MB attains a lower mean of 61.5 and a higher std of 5.8. We think the robustness of LA-MB could be attributed to its efficient hierarchical matching scheme, which avoids the ambiguity caused by the large matching space.

Feature-related Hyper-parameter We adopt a temperature $T$ to scale the values between the location and appearance features (see Eq. (5)). As shown in Tab. 5, our model is robust to the different values of $T$. We thus default its value to 1 for simplicity. Additionally, as shown in Tab. 6, our model is also robust to the different values of $\lambda$, which balances the weight between the stored appearance.
Table 4. Ablation study on stability of Video-kMaX using different memory-related hyper-parameter sets ($\tau$ for memory-refreshing and $\alpha$ for confidence threshold) on KITTI-STEP val set. We vary $\tau \in \{1, 3, 10, 20\}$ (different columns in the table) and $\alpha \in \{0.6, 0.7, 0.8\}$ (different rows in the table). We compute the mean and standard deviation column-wise (fixed $\tau$ and varied $\alpha$), row-wise (varied $\tau$ and fixed $\alpha$), and table-wise (varied $\tau$ and $\alpha$). We plot the mean and standard deviation for the whole table on the right. The proposed LA-MB is more robust to the hyper-parameter values than the naïve-MB approach. Our final LA-MB setting and the naïve-MB baseline are labeled with brown and red color, respectively.

Table 5. Ablation study on temperature $T$, which scales the values between location and appearance features. Our final setting is labeled with gray color. In this table, we show results up to two decimal points to more clearly see the robustness to $T$.

Table 6. Additional analysis on moving average weight $\lambda$, which balances the stored appearance feature in the memory and current appearance feature. Our final setting is labeled with gray color. In this table, we show results up to two decimal points to more clearly see the robustness to $\lambda$.

Visualization Analysis

We visualize results in Fig. 5 for KITTI-STEP. clip-kMaX performs better than the state-of-the-art TubeFormer [20] for consistent segmentation between frames in a clip. The proposed LA-MB enables long-term association, successfully re-identifying the occluded car object (ID 214), while the baseline naïve-MB fails, since it only exploits the appearance feature.

5. Conclusion

In this work, we proposed Video-kMaX, a unified framework for online and near-online Video Panoptic Segmentation (VPS) model with two modules: clip-kMaX and LA-MB. The clip-kMaX utilizes object queries as cluster centers to group pixels of the same object within a clip, while the LA-MB is a novel and robust memory module for both short- and long-term association with a hierarchical matching scheme. The effectiveness of our approach is demonstrated on the KITTI-STEP and VIPSeg datasets.

Acknowledgment

This work was supported by the NRF (NRF-2020M3H8A1115028, FY2021).
References


[26] Yanwei Li, Hengshuang Zhao, Xiaojuan Qi, Liwei Wang, Zeming Li, Jian Sun, and Jiaya Jia. Fully convolutional...


